



Universidad Carlos III de Madrid

Ingeniería Técnica en Informática de Gestión

*APLICACIÓN EDUCATIVA PARA APARATOS MÓVILES SOBRE*

*LOS RIESGOS INFANTILES*

**Julio 2011**

Alumno: Víctor Alonso Miranda

Tutor: Elena Castro Galán

Director: Fausto Sainz de Salces



## AGRADECIMIENTOS

---

Empezando cronológicamente, quiero agradecer a mi compañero Iván, todo el apoyo, ayuda e implicación que siempre ha mostrado para conseguir que obtuviera unos resultados óptimos en la carrera. Él ha sido, compañero de carrera, compañero de trabajo, amigo y vecino mío de toda la vida, y me alegra saber que hoy en día seguimos teniendo una gran relación y que ambos podemos contar el uno con el otro.

Agradecer a Elena y Fausto, toda la ayuda prestada a lo largo del proyecto. Ambos habéis colaborado activamente en mostrar vuestras mejores opiniones acerca del proyecto, me habéis ayudado a través de documentos y reuniones a realizar un juego lo más ameno y divertido posible, al igual que educativo, y me habéis mostrado gran ayuda para obtener una memoria lo mejor redactada posible.

Durante el desarrollo del proyecto, agradecer especialmente al pequeño Luismi, su gran ayuda, porque nunca pensé que un niño de tan solo 9 años pudiera aportarme tantas ideas de gran ayuda y me ayudara a entender cómo piensan y que es lo que les gusta a los niños de esa edad.

Agradecer con mucho cariño la participación activa en el desarrollo y dibujo de las pantallas a mi mujer Mara. Ella me ayudó a dibujar, a plasmar sobre una pantalla de móvil, esos dibujos sencillos, pero a la vez con contenido, que sirven para la educación de los más pequeños que puedan y quieran hacer uso del juego. Gracias por haber hecho tú, esas labores del hogar mientras yo pasaba el poco rato libre que me quedaba en la tarde para hacer el proyecto.

Y finalmente, el agradecimiento más especial a Enrique, Eulalia y Silvia, ellos son mis padres y hermana, porque ellos me han ayudado desde el primer día a conseguir este reto de finalizar los estudios, porque ellos me han apoyado siempre en la ilusión mutua de conseguirlo, y porque como todo aquel que vive bajo tu mismo techo, ellos han soportado las alegrías que produce aprobar un examen y la desilusión, nerviosismo y cambios de humor que supone suspenderlos después de haber llevado a cabo un largo tiempo de preparación. Hoy para mis padres también es un día especial y un gran motivo para sentirse orgullosos de ellos mismos y de sus hijos, pues vosotros nos disteis la oportunidad de realizar unos estudios universitarios y ambos hijos lo hemos aprovechado.

¡A todos vosotros muchísimas gracias!



AGRADECIMIENTOS .....	3
ÍNDICE DE CONTENIDOS .....	5
AGRADECIMIENTOS .....	3
ÍNDICE DE CONTENIDOS .....	5
ÍNDICE DE FIGURAS .....	8
ÍNDICE DE TABLAS .....	10
1. INTRODUCCIÓN .....	12
1.1 ESTRUCTURA DEL DOCUMENTO .....	12
1.2 IMPORTANCIA DEL JUEGO EDUCATIVO .....	13
1.3 CONTEXTO .....	14
1.4 OBJETIVOS Y BENEFICIOS .....	16
2. ESTADO DEL ARTE .....	18
2.1 TECNOLOGÍA JAVA .....	18
2.1.1 Lenguaje Programación Java .....	18
2.1.2 Tecnología J2ME .....	21
2.1.3 MIDLETs .....	23
2.2 ALMACENAMIENTO RMS .....	26
2.2.1 Introduccion .....	26
2.2.2 Almacén de Registros .....	27
2.2.3 Gestión del almacén .....	27
2.3 WIRELESS TOOLKIT .....	29
2.3.1 Introducción .....	29
2.3.2 Características y Beneficios .....	29
2.4 API DE JUEGOS DE MIDP2 .....	31
2.4.1 GameCanvas .....	31
2.4.2 Layer .....	33
2.4.3 TiledLayer .....	34
2.4.4. LayerManager .....	34
2.4.5 Sprite .....	35
3. INFORMACIÓN GENERAL DE SOFTWARE EDUCATIVO .....	37

3.1 DEFINICIÓN SOFTWARE EDUCATIVO .....	37
3.1.1. <i>Tipos de Software Educativo</i> .....	37
3.1.2. <i>Beneficios Software Educativo</i> .....	39
3.1.3. <i>Modelos de Aprendizaje</i> .....	40
3.1.4. <i>Evaluación de Software Educativo</i> .....	41
4. USABILIDAD .....	44
4.1. INTRODUCCIÓN .....	44
4.2. MÉTODOS .....	44
4.3. RESULTADOS .....	46
5. ACCESIBILIDAD .....	51
5.1 ACCESIBILIDAD EN LOS JUEGOS .....	52
5.1.1 <i>Diversidad Funcional Auditiva</i> .....	53
5.1.2 <i>Diversidad funcional cognitiva</i> .....	54
5.1.3 <i>Diversidad Funcional Visual</i> .....	54
5.1.4 <i>Diversidad Funcional en la Movilidad</i> .....	56
6. RIESGOS INFANTILES: INFORMACIÓN Y ACTUACIÓN.....	57
7. HISTORIA DEL JUEGO “UN DÍA CON TICO” .....	58
7.1 PRESENTACIÓN .....	58
7.2 DESARROLLO DEL JUEGO.....	59
8. DESARROLLO PROYECTO JUEGO EDUCATIVO PARA APLICACIÓN MÓVIL.....	66
8.1 FASE DE ANÁLISIS .....	66
8.1.1 <i>Descripción general del Sistema</i> .....	66
8.1.1.1 Capacidades generales.....	66
8.1.1.2 Diagramas de casos de uso .....	67
8.1.1.3 Restricciones generales .....	68
8.1.3 <i>Características de los Usuarios</i> .....	69
8.1.4 <i>Requisitos de Usuario</i> .....	69
8.1.4.1 Requisitos de Capacidad .....	70
8.1.4.2 Requisitos de Restricción .....	73
8.2 FASE DE DISEÑO.....	76
8.2.1 <i>Fecha de inicio y fecha de fin</i> .....	76
8.2.2. <i>Diagrama de Gantt</i> .....	77

Aplicación Educativa para aparatos móviles sobre los riesgos infantiles  
Víctor Alonso Miranda

8.2.3 Estimación de Costes .....	77
8.2.4 Diagrama de Clases.....	80
8.2.5 Diagrama de Secuencias.....	81
9. VALIDACIÓN.....	83
10. CONCLUSIONES Y LÍNEAS FUTURAS .....	85
GLOSARIO.....	88
BIBLIOGRAFÍA.....	90
ANEXO.....	92

Imagen 1: Ciclo Vida Midlet .....	24
Imagen 2: Estados Midlet .....	24
Imagen 3: Resultados 1 .....	46
Imagen 4: Resultados 2 .....	47
Imagen 5: Resultados 3 .....	48
Imagen 6: Resultados 4 .....	49
Imagen 7: Resultados 5 .....	50
Imagen 8: Pantalla Inicial .....	59
Imagen 9: El Despertar .....	60
Imagen 10: El Desayuno .....	60
Imagen 11a: Bonus I Dibujo .....	61
Imagen 11b: Bonus I Pregunta .....	61
Imagen 12a: Bonus II Dibujo .....	61
Imagen 12b: Bonus II Pregunta .....	61
Imagen 13a: Bonus III Dibujo .....	61
Imagen 13b: Bonus III Pregunta .....	61
Imagen 14a: Bonus IV Dibujo .....	62
Imagen 14b: Bonus IV Pregunta .....	62
Imagen 15: Preparamos el Baño .....	62
Imagen 16: La Ducha .....	63
Imagen 17: Cristales Rotos .....	63



Aplicación Educativa para aparatos móviles sobre los riesgos infantiles  
Víctor Alonso Miranda

Imagen 18: Incendio Cocina .....	64
Imagen 19: Elección animal Ranking .....	64
Imagen 20: Datos animal y puntos .....	65
Imagen 21: Ranking .....	65
Imagen 22: Ayuda .....	65
Imagen 23: Diagrama Casos de Uso .....	68
Imagen 24: Diagrama Gantt .....	77
Imagen 25: Diagrama de Clases .....	80
Imagen 26: Diagrama de Secuencias .....	82
Imagen 27. Crear paquete Jar .....	93
Imagen 28. Archivos Jar .....	94
Imagen 29. Menú .....	96
Imagen 30. Pantallas I .....	98
Imagen 31. Pantallas II .....	100
Imagen 32. Pantallas III .....	101
Imagen 33. Elección animal Puntuación .....	103
Imagen 34. Datos Puntuación .....	104
Imagen 35. Ranking .....	105
Imagen 36. Ayuda .....	106

Tabla 1: Tipos de Software Educativo.....	38
Tabla 2: RU-C-01 .....	70
Tabla 3: RU-C-02.....	71
Tabla 4: RU-C-03 .....	71
Tabla 5: RU-C-04 .....	72
Tabla 6: RU-C-05 .....	72
Tabla 7: RU-R-01 .....	73
Tabla 8: RU-R-02 .....	73
Tabla 9: RU-R-03 .....	74
Tabla 10: RU-R-04 .....	74
Tabla 11: RU-R-05 .....	75
Tabla 12: RU-R-06 .....	75
Tabla 13: RU-R-07.....	76
Tabla 14: Recursos Proyecto .....	78
Tabla 15: Costes Proyecto .....	79



# 1. INTRODUCCIÓN

---

## 1.1 ESTRUCTURA DEL DOCUMENTO

\* **Capítulo 1:** En este primer capítulo, se realiza una introducción de los motivos de la aplicación, así como una breve información sobre los beneficios que aportan los juegos educativos en el aprendizaje de los niños, comentando los objetivos que se trata de conseguir a través de este tipo de juegos

\* **Capítulo 2:** Se desarrolla una explicación del Estado del Arte del proyecto mostrando una explicación de la tecnología utilizada, así como del uso de la base de datos que se utiliza. Por otra parte se realiza una breve explicación del simulador utilizado y se nombre la API utilizada

\* **Capítulo 3:** Contiene una explicación de lo que es hoy en día el Software educativo indicando los diferentes tipos de Software Educativo existentes y los beneficios que producen. También muestra información sobre los diferentes modelos de aprendizaje existentes y habla sobre las formas de evaluar los resultados de emplear un software educativo.

\* **Capítulo 4:** En este capítulo se relata una breve explicación sobre qué métodos utilizar para conocer que piensan los usuarios sobre la aplicación y en consecuencia se muestran los resultados de una pequeña encuesta y además una breve evaluación de esos resultados.

\* **Capítulo 5:** Se muestra una breve explicación sobre la accesibilidad en los juegos educativos y se explica las diferentes disfunciones sensoriales existentes intentando mostrar posibles soluciones para que cualquier persona con una disfunción pueda hacer uso a los juegos educativos.

\* **Capítulo 6:** Breve explicación de los riesgos infantiles que pueden sufrir los pequeños a los que va destinada la aplicación en los hogares.

\* **Capítulo 7:** Contiene una breve explicación de cómo se desarrollará el juego. Una explicación del desarrollo de cada pantalla de nuestro juego con el objetivo de que los más pequeños consigan asimilar la forma de actuación ante los diferentes riesgos y desarrollando también alguna pantalla que permita al usuario a través de su habilidad con el móvil, aumentar su puntuación, para mayor disfrute del joven usuario

\* **Capítulo 8:** En este capítulo desarrollamos la Fase de Análisis y de Diseño del proyecto. Por una parte explicamos los modelos de Usuario a los que dirigimos la aplicación. El

modelo de usuario representa las características necesarias de dicho usuario en relación al contexto de la aplicación, teniendo en cuenta las habilidades físicas y perceptivas, las destrezas para identificar y realizar tareas, o la motivación, entre otras cosas. También exponemos las restricciones y el diagrama de Casos de Uso.

Por otra parte, en este capítulo, en la fase de diseño, exponemos el Diagrama de Gantt, el Diagrama de Clases, el Diagrama de Secuencias, y de igual forma exponemos la relación de fecha de inicio y fin del proyecto, así como la estimación de costes.

\* **Capítulo 9:** Se detalla una explicación de las múltiples validaciones que se han llevado a cabo para verificar y cerciorarse del correcto funcionamiento de la aplicación.

\* **Capítulo 10:** En este último capítulo se detallan las diversas conclusiones obtenidas tras la realización de este proyecto.

## 1.2 IMPORTANCIA DEL JUEGO EDUCATIVO

El juego educativo lo podemos definir como una actividad humana en general y particularmente infantil. Muchos autores han definido lo que es el juego y todos ellos coinciden en que *el juego contiene un valor funcional y una gran importancia para el desarrollo y crecimiento del ser humano*. (Jiménez, E.,2006)

*El juego es libre y espontáneo y el individuo siempre es consciente de que está desarrollando una actividad libre con la que dispone de un espacio personal con margen de error.*

*Además a través del juego, podemos desarrollar un modo de expresión que nos permite expresar sentimientos, motivaciones, tendencias, actitudes....*

*Lo más importante es que el juego es una actividad vivencial, donde el niño es un ser único que expresa sus actitudes, deseos, creencias y capacidades relacionadas con el desarrollo físico, afectivo y cognitivo.*

*El juego es una forma de interactuar con la realidad, propio de la infancia, caracterizado por la universalidad, regularidad y consistencia, sabiendo, que se desarrolla dentro de un marco psicológico que evoluciona con la edad, reflejando siempre la forma en que el niño concibe el mundo*

*Gracias al juego también facilitamos el desarrollo afectivo del niño, ya que el juego, actúa como un lenguaje privilegiado para el niño con el cual expresa los rasgos de su personalidad.*

*Otras funciones de los juegos, es que sirven de entretenimiento para los niños, con lo cual, conseguimos que no estén ociosos y les despierta el sentimiento de libertad.*

*Los contenidos de los juegos, están principalmente relacionados con la vida, con el trabajo y con la actividad de los adultos en la sociedad, lo que hace, que podamos utilizarlos como medio integrador para los niños en sus respectivas fases de la vida, intentando siempre que adquieran un determinado conocimiento..*

### **1.3 CONTEXTO**

El juego educativo debemos concebirlo como una actividad cuyo objetivo es que el usuario asimile determinados conocimientos. Normalmente están destinados a los niños, para que aprendan determinadas cosas sobre una materia o bien para que practiquen una determinada actividad. Tratándose de un juego, estos suelen ser divertidos y se trata de hacerlos lo más ameno posible para que cuanto más entretengan, más afiancen en el niño los conocimientos que se quiere que se asimilen.

Respecto a los juegos electrónicos educativos, debemos saber que son sistemas de enseñanza en los que se mezcla por un lado una o varias actividades pedagógicas y por el otro lado existen entornos gráficos que siempre se debe tratar que sean lo más divertido posibles intentando aumentar la motivación del alumno a la hora de desarrollar el juego.

Analizando 68 estudios desde 1963 hasta 1991 sobre Ciencias Sociales, Matemáticas, Lengua, Lógica, Física y Biología, encontraron que los juegos eran más efectivos en Matemáticas y en Lengua.(Carmona, C et al., 2006)

Otro punto importante a tener en cuenta es que el juego educativo debe adaptar las actividades que propone al nivel cognitivo del usuario, es decir debe de acometer unos objetivos para poder ser realmente educativo. Piaget desarrolló la Teoría del Desarrollo Cognitivo donde establece una definición precisa y amplia de los estados cognitivos de los niños. Las habilidades cognitivas que estudió Piaget en cada una de las etapas son muy importantes en el aprendizaje porque ayudan a determinar qué es lo que hay que enseñar y en qué momento del desarrollo se debe hacer y por supuesto, cual es la mejor forma de ponerlo en práctica. No es fácil encontrar sistemas educativos basados en los niveles cognitivos de Piaget.

En cuanto a la definición de “calidad” de un juego educativo, según diversos autores, se debe cumplir lo siguiente: (Carmona, C et al., 2006)

*1) Como en todos los materiales para la instrucción, los juegos educativos deben ser apropiados, es decir, accesibles para el nivel de desarrollo del usuario al que está dirigido;*

*2) Los estudiantes deben superar muchas barreras para realmente divertirse, como puede ser el miedo al fallo, el miedo al desconcierto y el temor a perder el control;*

*3) Los juegos son divertidos gracias a la interacción. Los jugadores son los responsables de sus decisiones porque estas decisiones influyen en el resultado. El éxito en un juego se puede reflejar en el esfuerzo y la habilidad del usuario;*

*4) En los buenos juegos el usuario tiene libertad de movimientos. En este punto es necesario indicar que esta libertad debe estar “supervisada” por técnicas que eviten la confusión y la desorientación;*

*5) Si se utilizan refuerzos, el usuario repetirá aquellos comportamientos que son recompensados y abandonará aquellos que son ignorados o penalizados;*

*6) Un buen juego es aquél que los niños quieran jugar una y otra vez. Para ello deben tener unas reglas y objetivos claros; debe ser fácil que los usuarios sepan cuál es su progreso a medida que van jugando; se debe poder utilizar varias estrategias; el juego debe ser tan motivador que hace que los niños mantengan el interés al enfrentarse a nuevos retos y a mejorar sus estrategias para ser mejores jugadores. Aunque muchos de los juegos actuales cuentan con sonidos, gráficos y otros efectos especiales que captan la atención del jugador, si el juego no está bien estructurado, estos elementos pueden resultar molestos. Para que el juego sea educativo, además de todo lo anterior, el juego debe proporcionar un aprendizaje.*

*7) En el caso de que el niño se quede estancado, un buen juego debe proporcionarle ayuda inmediata para evitar la frustración (normalmente a través de pistas).*

Resumiendo, las principales características de un buen juego educativo son:

Accesibilidad, adaptación, aprendizaje eficaz, ayuda, bien estructurado, desafío, inmediatez, interacción, libertad de navegación, motivación, orientación, privacidad, refuerzo, resultados, simplicidad y variedad.

#### 1.4 OBJETIVOS Y BENEFICIOS

El objetivo principal de los juegos educativos, es, como su propia descripción nos indica educar a los más pequeños, proporcionar un aprendizaje más dinámico y divertido a través del juego de una determinada materia proporcionando al alumno diferentes beneficios. A parte de los niños, los juegos educativos, pueden utilizarse como método de aprendizaje en personas con deficiencias psíquicas, puesto que de igual forma, para ellos también es una manera más fácil de adquirir nuevos conocimientos.

Un niño puede aprender a manejar un ratón con tan solo 2 años y nos sorprendería el saber lo poco que tardan en aprender a manejar un ordenador.

Los principales beneficios que proporcionan los juegos educativos en un niño se pueden describir de la siguiente forma: (Compute-rs.com, 2011)

- 1) *Lo principal de cara a un futuro más lejano, es que el niño aprenderá a manejar un ordenador.*
- 2) *Más acorde con los propios beneficios de los juegos como tal, vemos que éstos, mejoran la coordinación mano – ojo, así como la motricidad fina en el más pequeño.*
- 3) *Permiten que se desarrollen habilidades que les ayuda a la hora de la toma de decisiones.*
- 4) *Mejoran el rendimiento escolar del niño, ayudándolo principalmente en materias como las matemáticas o lenguaje.*
- 5) *Por otra parte, a través de los juegos educativos, el niño aprende a resolver problemas, como en el caso que estudiamos, intentando enseñar al niño como actuar ante determinadas situaciones de emergencia. También ayudan al niño a establecer metas y otras habilidades cognitivas importantes. Esto es debido a que en la mayoría de los juegos aumenta su dificultad con la progresión a los niveles superiores, por lo tanto hace que el reto del siguiente nivel sea más emocionante y le dé al niño un sentido de logro y satisfacción personal con la finalización de cada nivel.*

Los juegos de ordenador, móvil o consola pueden producir muchos beneficios en el desarrollo de un niño pero sin embargo, el niño no debe pasar todo su tiempo delante del ordenador. El ejercicio y principalmente jugar e interactuar con los demás niños es muy importantes en el desarrollo de un niño.



Aplicación Educativa para aparatos móviles sobre los riesgos infantiles  
Víctor Alonso Miranda

De esta forma, este proyecto tiene un objetivo principal:

\*        Desarrollar un juego educativo para móviles que facilite el conocimiento acerca de los riesgos cotidianos que le pueden surgir a un niño en el entorno de su hogar y como actuar ante ellos.

## 2. ESTADO DEL ARTE

---

En este capítulo se hace un repaso de las tecnologías utilizadas en este proyecto. En primer lugar se presentará el lenguaje de programación Java, en concreto la plataforma J2ME, utilizada para el desarrollo de aplicaciones móviles.

Se desarrolla una explicación sobre el emulador de juegos y aplicaciones para móviles Wireless Toolkit, el cual permite visualizar y manipular el juego durante su programación.

### 2.1 TECNOLOGÍA JAVA

#### 2.1.1 Lenguaje Programación Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. (Gosling, J., 2012)

Las aplicaciones Java son típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995.

Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través de Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre.

Las dos principales características del lenguaje Java, es que es un lenguaje orientado a objetos y que es independiente de la plataforma.

La primera característica, *orientado a objetos* (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos.

Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos.

Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode)—instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT.

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCJ. Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas, aunque sigue siendo mucho más lento que otros lenguajes.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad. Otra técnica, conocida como compilación JIT (Just In Time, o “compilación al vuelo”), convierte el bytecode a código nativo cuando se ejecuta la aplicación. Otras máquinas virtuales más sofisticadas usan una “recompilación dinámica” en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad en ambos.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura.

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

### **2.1.2 Tecnología J2ME**

Al principio de los 90, Sun Microsystems creó un nuevo lenguaje de programación llamado Oak como parte de un proyecto de investigación para construir productos electrónicos que dependan principalmente del software. (CiberAula Java, 2010)

El primer prototipo para Oak fue un controlador portable llamado Star7, un pequeño dispositivo handheld con una pantalla touchscreen LCD que tenía incorporado soporte a redes inalámbricas y comunicaciones infrarrojas. Este dispositivo podría ser usado como control remoto para televisores o VCR y como guía de programas electrónicos, e incluso tenía algunas funciones que ahora son asociadas a los PDAs, como agenda de citas.

El software para este tipo de dispositivos necesitaba ser extremadamente confiable y no debía hacer excesivo uso de memoria ni requerir demasiada potencia en el procesador. Oak fue desarrollado como resultado de la experiencia del equipo de desarrollo con el lenguaje C++, el cual, a pesar de tener muchas grandes características, demostró que era un lenguaje complejo y ocasionaba que los programadores comentan fácilmente errores y eso afectaba la confiabilidad del software.

Oak fue diseñado para quitar o reducir la posibilidad de que los programadores comentan errores, ¿cómo? detectando la mayoría de errores en tiempo de compilación y quitando algunas de las características del lenguaje C++ (como punteros y la administración de memoria controlada por el programador) que eran los problemas más comunes.

Desafortunadamente, el mercado para el tipo de dispositivos que el nuevo lenguaje fue creado no se desarrolló tanto como Sun Microsystems esperaba, y al final ningún dispositivo basado en Oak fue vendido a los clientes. Sin embargo, al mismo tiempo, el inicio del conocimiento público de Internet produjo un mercado para el software de navegación para Internet (los navegadores Web). En respuesta a esto, Sun Microsystems renombró el lenguaje de programación Oak a Java y lo usó para desarrollar un navegador multiplataforma llamado

## Aplicación Educativa para aparatos móviles sobre los riesgos infantiles

Víctor Alonso Miranda

HotJava. También le dio la licencia de Java a Netscape, quienes lo incorporaron en su navegador que por ese entonces era el más popular en el mercado, luego fueron incorporados los Java applets.

En un par de años, las capacidades multiplataforma del lenguaje de programación Java y su potencia como plataforma de desarrollo para aplicaciones que podían ser escritas una vez y ejecutadas en diversos sistemas Windows y Unix, había despertado el interés de usuarios finales, porque vieron en ella una manera de reducir los costos del desarrollo de software.

Con el objetivo de conocer las necesidades de los experimentados desarrolladores en Windows y Motif/X-Windows para crear aplicaciones para usuarios finales sofisticados acostumbrados a usar interfaces ricas, Sun Microsystems rápidamente expandió el alcance y tamaño de la plataforma Java. Esta plataforma extendida incluyó un conjunto más complejo de librerías de interfaces de usuario que aquellos que usaran para construir applets, además con un conjunto de características de computación distribuida y seguridad mejorada.

Con el tiempo Sun Microsystems liberó la primera versión de la plataforma Java 2, había sido necesario dividirla en varias piezas. La funcionalidad principal, estimado como el mínimo soporte requerido para cualquier ambiente Java, estaba empaquetada en el Java 2 Standard Edition (J2SE).

Muchos paquetes opcionales pueden ser agregados al J2SE para satisfacer requerimientos específicos para aplicaciones particulares, como extensiones seguras de sockets que permitan el comercio electrónico. Sun Microsystems también respondió al incremento del interés de usar Java para el desarrollo a un nivel empresarial, y ambientes de servidores de aplicaciones con la plataforma Java 2 Enterprise Edition (J2EE), el cual incorpora nuevas tecnologías como servlets, Enterprise JavaBeans, JavaServer pages, etc.

Como la mayoría de software, los requerimientos de recursos de Java tienen un incremento con cada nueva versión que aparece. A pesar que Java tiene sus raíces en el software para productos electrónicos pequeños, J2SE requiere mucha más memoria y potencia en el procesador para que sea una solución viable en el mercado.

Irónicamente, mientras Sun Microsystems estaba desarrollando Java para Internet y para la programación comercial, la demanda empezó a crecer en los dispositivos pequeños e incluso en tarjetas inteligentes, retornando Java a sus raíces.

Sun Microsystems respondió a esta demanda creando varias plataformas Java con funcionalidades reducidas, cada una hecha a la medida de un segmento vertical y específico del mercado.

Estas plataformas reducidas están todas basadas en el JDK 1.1, el predecesor de la plataforma Java 2, y cada una tiene una estrategia diferente al problema de reducir la plataforma para acomodarla a los recursos disponibles. Por lo tanto, cada una de estas plataformas de funcionalidad reducida representan una solución ad hoc al problema. Por ello es que aparece la plataforma J2ME, para reemplazar todas esas plataformas reducidas basadas en el JDK 1.1 y crear una sola solución basada en Java 2.

En conclusión, J2ME es la versión de Java orientada a los dispositivos móviles. Debido a que los dispositivos móviles tienen una potencia de cálculo baja e interfaces de usuario pobres, es necesaria una versión específica de Java destinada a estos dispositivos, ya que el resto de versiones de Java, J2SE o J2EE, no encajan dentro de este esquema. J2ME es por tanto, una versión “reducida” de J2SE.

### 2.1.3 MIDLETs

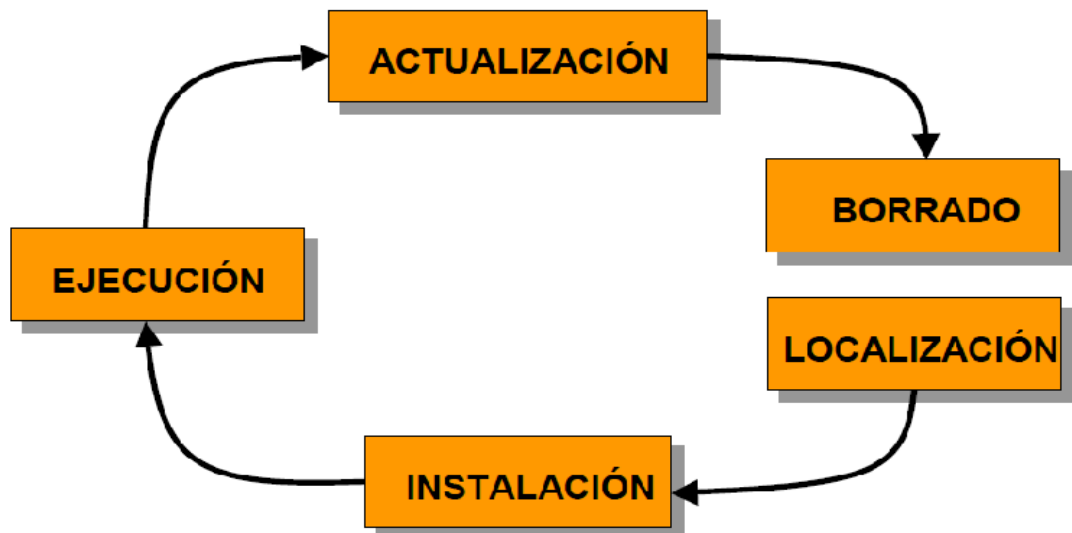
Los **midlets** son aplicaciones creadas usando la especificación **MIDP**. Están diseñados para ser ejecutados en dispositivos con poca capacidad gráfica, de cómputo y de memoria.(García Galán,S.)

Estos dispositivos no disponen de una línea de comandos donde poder ejecutar las aplicaciones, si no que reside en él un software que es el encargado de ejecutar los midlets y gestionar los recursos que éstos ocupan, **el gestor de aplicaciones**.

El gestor de aplicaciones (Application Management System) realiza dos grandes funciones:

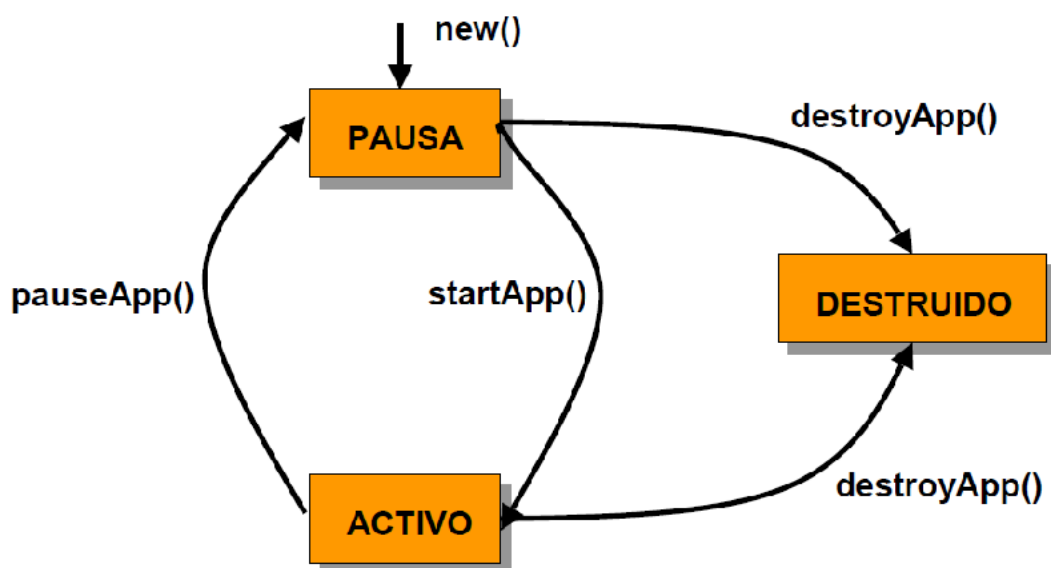
- \* Gestiona el ciclo de vida de los midlets.
- \* Controla los estados por los que pasa el midlet mientras está en la memoria del dispositivo.

El ciclo de vida de un Midlet está compuesto por:



*Imagen 1 Ciclo vida Midlet*

Los diferentes estados por los que puede pasar un Midlet son:



*Imagen 2 Estados Midlet*

El paquete **javax.microedition.midlet** define las aplicaciones MIDP y su comportamiento con respecto al entorno de ejecución

**MIDlet** : Aplicación MIDP



**MIDletStateChangeException** : Indica que el cambio de estado ha fallado.

Las diferentes clases que forman el **Paquete javax.microedition.midlet** son:

**protected MIDlet()**: Constructor de la clase, si la llamada a este constructor falla se lanzaría la excepción *SecurityException*.

**public final String getAppProperty(String key)**: Permite recuperar el valor de las propiedades desde el AMS. Las propiedades se consiguen por medio de los archivos *manifest* y *JAD*. El nombre de la propiedad a recuperar debe indicarse en el parámetro *key*. El método nos devuelve un string con el valor de la propiedad o null si no existe ningún valor asociado al parámetro *key*. Si *key* es null se lanzará la excepción *NullPointerException*.

**public final void notifyDestroyed()**: Se utiliza para indicar al AMS que el midlet ha entrado en el estado de destruido. El AMS considera que los recursos ocupados por el midlet están libres.

**public final void notifyPaused()**: Se notifica al AMS que el midlet no quiere estar activo y que ha entrado en el estado de pausa. Sólo debe invocarse este método cuando el midlet esté activo. El midlet puede volver al estado de activo invocando el método *MIDlet.startApp()* o ser destruido llamando al método *MIDlet.destroyApp()*. Si la aplicación es pausada por si misma, es necesario llamar al método *MIDlet.resumeRequest()* para volver al estado “Activo”.

**public final void resumeRequest()**: Proporciona un mecanismo a los midlets para indicar al AMS su interés de pasar al estado “Activo”, por lo tanto será el AMS quién determine qué aplicaciones pasarán a dicho estado invocando el método *MIDlet.startApp()*.

**public final int checkPermission(String permission)**: Consigue el estado del permiso especificado en el atributo *MIDlet-Permission* del archivo *JAD*. Los valores devueltos son:

- \* 0 si el permiso es denegado o no existe.
- \* 1 si el permiso es permitido
- \* -1 si el estado es desconocido

**protected abstract void pausedApp()**: Indica al midlet que entre en el estado de “Pausa”. Sólo debe ser invocado cuando se esté en el estado de “Activo”. Si ocurre una excepción

*RuntimeException* durante la llamada `MIDlet.pauseApp()`, el midlet será destruido inmediatamente. Se llamará a su método `MIDlet.destroyApp()` para liberar recursos.

***protected abstract void startApp() throws MIDletStateChangeException:*** Indica al midlet que ha entrado en el estado “Activo”. Sólo se puede invocar cuando el midlet está en el estado de “Pausa”. Si en ese momento no puede pasar al estado “Activo” se lanzará la excepción *MIDletStateChangeException*.

***protected abstract void destroyApp(boolean unconditional) throws MIDletStateChangeException:*** Indica la terminación del midlet y su paso al estado “Destruído”. El midlet debe liberar todos los recursos y salvar cualquier data en el almacenamiento persistente que deba ser guardado. Este método puede ser llamado desde los estados “Pausa” y “Activo”. Si el parámetro *unconditional* es *false*, el midlet puede lanzar la excepción *MIDletStateChangeException* para indicar que no puede ser destruido en ese momento. Si es *true*, el midlet asume el estado de destruido independientemente de cómo finalice el método.

Por último nombrar otra clase importante para los Midlets la cual es ***MIDletStateChangeException*** y es lanzada por los eventos `destroyApp()` y `startApp()`.

## 2.2 ALMACENAMIENTO RMS

### 2.2.1 Introducción

Un dispositivo móvil no dispone de disco duro donde almacenar información permanente. J2ME resuelve el problema mediante el RMS (Record Management System). RMS es un pequeño sistema de base de datos muy sencillo, pero que nos permite añadir información en una memoria no volátil del móvil. RMS no tiene nada que ver con JDBC debido a las limitaciones de los dispositivos J2ME, por lo tanto, el acceso y almacenamiento de la información se hace a mucho más bajo nivel. RMS no puede ser consultado con sentencias SQL, ni nada parecido. En una base de datos RMS, el elemento básico es el registro (record). Un registro es la unidad de información más pequeña que puede ser almacenada. Los registros son almacenados en un `recordStore` que puede visualizarse como una colección de registros. Cuando almacenados un registro en el `recordStore`, a éste se le asigna un identificador único que identifica unívocamente al registro.

Para poder utilizar RMS hemos de importar el paquete *javax.microedition.ms*. Este paquete nos provee de la clase *RecordStore* y de cuatro interfaces *RecordComparator*, *RecordEnumeration*, *RecordFilter*, *RecordListener*

### 2.2.2 Almacén de Registros

Un almacén de registros (*recordStore*) consiste en una colección de registros que persistirán a lo largo de múltiples invocaciones de un MIDlet. Esos almacenes de registros se crean en zonas dependientes de la plataforma que no está dispuesto directamente a los MIDlets. El espacio de nombres del almacén de registro se controla en el ámbito del MIDlet Suit y sólo está permitida la manipulación de los almacenes de registros a las suites de MIDlets que los poseen, por otra parte no existe mecanismo para compartir registros entre MIDlets que pertenezcan a suites diferentes.

Los nombres de los almacenes de registros es sensible a las mayúsculas y puede consistir en cualquier combinación de hasta 32 caracteres Unicode.

No se suministran operaciones de bloqueo y la implementación de los almacenes de registros asegura que todas las operaciones sobre los mismos son atómicas, síncronas y serializadas, por lo que no puede llegar a ocurrir que se corrompan con accesos múltiples.

Por otra parte, el almacén guarda el día y hora de la última modificación y mantiene una versión (cada vez que hay un cambio se incrementa ésta). Esto es útil para sincronización de aplicaciones. Debemos saber también que cuando una aplicación se elimina del dispositivo también se elimina los almacenes que tuviera.

### 2.2.3 Gestión del almacén

Antes de poder almacenar un registro hemos de abrir un *recordStore* con el método *openRecordStore*.

***static RecordStore openRecordStore(String nombre, boolean crear)***: Abre (y en su caso crea) un almacén de registros asociado con la correspondiente suite MIDlet. Las excepciones que nos puede provocar este método son: *RecordStoreException*, *RecordStoreFullException*, *RecordStoreNotFoundException*.

*Nombre* es el parámetro que nos indica el nombre de la base de datos. Puede tener un tamaño de 32 caracteres. El parámetro *crear* si tiene su valor a *true*, creará la base de datos si no existe.

Cuando terminemos de utilizar el recordStore, hemos de cerrarlo:

***RecordStore.closeRecordStore():*** Un almacén de registros debería cerrarse tan pronto como se acabe de usar, ya que consume recursos, pero debemos saber que el método `closeRecordStore()` realmente no cierra el almacén, sino que informa que el proceso o hilo ha dejado de usarlo. El almacén se cierra sólo cuando todos los procesos / hilos que lo utilizaban han llamado al método de cierre.

El borrado de registros se realiza con el método `deleteRecord()`:

***Public void deleteRecord( int recordId):*** A través del parametro *recordId* indicamos que registro queremos borrar de nuestra base de datos. Además, al igual que en escritura y en la lectura de registros hemos de tener en cuenta que puede provocar la excepción *RecordStoreException*. Debemos saber que se borra el registro indicado pero no se libera espacio, sino que se añade a la lista de registros libres para su posterior uso. Los antiguos

*RecordIds* no volverán a ser utilizados

La lectura de los registros se realizará con el método `getRecord()`:

***Public byte[ ] getRecord(int Id):*** El método `getRecord()` nos permite acceder al registro que deseemos, siempre que conozcamos su identificador. No es necesario que almacenemos y mantengamos una lista con todos los identificadores de los registros debido a que existe un metodo llamado *recordEnumeration* que nos permite conocer el identificador de cada registro. Con el método `getRecord()` debemos capturar igualmente la excepción *RecordStoreException*.

Como anteriormente hemos comentado, el método `enumerateRecords()` es el destinado para recorrer los registros.

***Public RecordEnumeration enumerateRecords(RecordFilter filtro, RecordComparator comparador, boolean Actualizar):*** Los dos primeros parámetros sirven para personalizar el recorrido de los registros. Gracias al primero podemos filtrar la búsqueda y con el segundo podemos recorrer los registros de forma ordenada. El parámetro *actualizar* indica si la enumeración debe actualizarse cuando realicemos alguna operación de inserción o borrado de registros. Si se va a realizar un recorrido rápido por los registros es mejor ponerlo a *false* para evitar la sobrecarga.

El *RecordEnumeration* tiene dos métodos interesantes, por una parte tiene el *hasNextElement()* que devolverá el valor *true* si hay un siguiente elemento disponible para ser leído. Cada vez que leemos un elemento se adelanta al puntero siguiente. Por otra parte contiene el método *nextRecordId()* que nos devuelve el identificador del siguiente registro.

## **2.3 WIRELESS TOOLKIT**

### **2.3.1 Introducción**

El Sun Java Wireless Toolkit para CLDC es una caja de herramientas para el desarrollo de aplicaciones inalámbricas que se basan en Java ME. Está conectado con Configuración Limitada de Dispositivos (CLDC) y Mobile Information Device Profile (MIDP), y diseñado para funcionar en teléfonos móviles, asistentes digitales personales convencionales y otros dispositivos móviles pequeños. El kit de herramientas incluye los entornos de emulación, la optimización del rendimiento y las características de ajuste, documentación y ejemplos que los desarrolladores necesitan para llevar las aplicaciones inalámbricas eficientes y exitosas al mercado rápidamente.(Oracle, 2012)

### **2.3.2 Características y Beneficios**

El Sun Java Wireless Toolkit incluye muchas características amigables para desarrolladores:

- \* Apoyo a la plataforma API siguientes:
- \* La tecnología Java para la industria inalámbrica (JTWI)
- \* Mobile Service Architecture (MSA)
- \* Apoyo a la siguiente API estándar:
  - Connected Limited Device Configuration (CLDC)
  - Mobile Information Device Profile (MIDP) 2.0
  - API de mensajería inalámbrica (WMA) 2.0
  - Mobile Media API (MMAPI)
  - PDA paquetes opcionales para la plataforma J2ME
  - API Java para Bluetooth

- J2ME Web Services Specification

- Mobile 3D Graphics API para J2ME

- \* Seguridad y confianza API de servicios (SATSA)
- \* Situación de la API (LAPI) 1.0
- \* Contenido Handler API (CHAPI) 1.0
- \* Protocolo iniciado sesión (SIP) de la API
- \* Escalable de gráficos vectoriales 2D (SVG) de la API
- \* El pago de la API (PAPI)
- \* Advanced Multimedia Supplements (AMMS)
- \* Mobile Internacionalización de la API
- \* Enlaces para Java API de OpenGL
- \* Elección de pieles emulador y características mejoradas de la pantalla
- \* Crear un proyecto de Jar y jad
- \* Apoyo a terceros emuladores
- \* La supervisión del rendimiento mejorado
- \* Aplicaciones de demostración
- \* Personalización
- \* Integración con los populares entornos de desarrollo integrados (IDE)

El Sun Java Wireless Toolkit se puede utilizar como un entorno de desarrollo independiente o con un IDE, como el de NetBeans Mobility Pack.

El kit de herramientas de emulador cumple plenamente con los correspondientes equipos de la API de compatibilidad de la tecnología, asegurándose de que todas las API están presentes y van a reaccionar en consonancia con las implementaciones compatibles. En el modo

independiente, los usuarios pueden configurar las preferencias individuales, crear aplicaciones, crear Java Archive (JAR) y Java Application Descriptor (JAD) archivos, y más, utilizando una interfaz amigable el conjunto de herramientas de KToolbar, o su línea de comandos. Cuando se integra con un IDE, las utilidades de la caja de herramientas y las preferencias aparecen en las selecciones de menú del IDE, y también se puede controlar desde el IDE de interfaz de línea de comandos, además soporta el conjunto de herramientas de depuración a nivel de fuente.

## **2.4 API DE JUEGOS DE MIDP2**

La versión 2.0 del perfil MID (MIDP 2.0) nos ofrece cinco nuevas clases que nos facilita la programación de juegos o aplicaciones gráficas en J2ME. Sin embargo, hay que tener en cuenta que no todos los dispositivos actuales soportan esta nueva versión de MIDP. (García Serrano,A.)

Las nuevas clases de las que disponemos son:

***GameCanvas***

***Layer***

***LayerManager***

***TiledLayer***

***Sprite***

### **2.4.1 GameCanvas**

La clase GameCanvas es una subclase de Canvas que añade nuevas capacidades orientadas a la programación gráfica. Básicamente nos da la posibilidad de dibujar de forma rápida e inmediata (sin parpadeos) en el Canvas, automatizando el uso del doble buffer .

Además nos permite el acceso al estado de las teclas del dispositivo. Nos va a

permitir incluir dentro del game loop toda la lógica de repintado y acceso al teclado, pudiendo así implementar todo el game loop en el mismo thread de ejecución.

La lógica de juego, el dibujado de los gráficos y la gestión del teclado se realizan en métodos diferentes. Además, el game loop tiene que implementarse en un thread distinto al que se encarga de responder a los eventos de teclado y a las peticiones de repintado de la pantalla. Esto es bastante ineficiente y perdemos mucho control sobre el game loop. Por ejemplo, cuando

salta un evento de repintado nuestro game loop puede estar en mitad de un proceso de cálculo y dibujar el escenario de forma inconsistente.

Con GameCanvas podemos implementar el repintado y la lectura del teclado dentro del propio game loop, lo que quiere decir que correrán en el mismo hilo de ejecución y tendremos el control total sobre cuando queremos hacer el repintado o leer el teclado.

La clase GameCanvas debe ser heredada, ésta a su vez es una clase hija de Canvas. Por lo tanto te permite seguir utilizando los métodos y constantes de la clase Canvas. Además GameCanvas utiliza de forma transparente un doble buffer o offscreen. Cuando pintamos sobre el GameCanvas, en realidad no estamos pintando sobre la pantalla del dispositivo, sino que lo hacemos sobre una zona de memoria que representa una pantalla virtual. Una vez que hemos finalizado todo el proceso de pintado, sólo nos resta decirle al GameCanvas que vuelque la pantalla virtual sobre la pantalla real del dispositivo. Esto lo hacemos con el método `flushGraphics()`.

Los métodos que añade GameCanvas a Canvas son los siguientes.

***getGraphics():*** Devuelve un objeto de tipo Graphics sobre el que podemos realizar todas las operaciones de pintado. Realmente este objeto Graphics representa la pantalla virtual o offscreen del doble buffer.

***flushGraphics():*** Vuelca la pantalla virtual o offscreen en la pantalla real del dispositivo.

***flushGraphics(int x, int y, int ancho, int alto):*** Vuelca una región de la pantalla virtual o offscreen en la pantalla real del dispositivo. La región

se especifica mediante la posición x e y de la esquina superior izquierda y el ancho y el alto de ésta.

***paint(Graphics g):*** Pinta el GameCanvas en el offscreen.

***getKeyStates():*** Devuelve un entero (int) que nos dice que teclas están pulsadas (o han sido pulsadas desde la última llamada a `getKeyStates()`) y cuáles no. Las teclas se codifican en el entero de forma que a cada una le corresponde un bit. Si la tecla está pulsada o fue pulsada desde la última invocación a `getKeyStates()` su correspondiente bit estará a 1. Si no, su valor será 0.

Para facilitarnos la lectura de este valor, GameCanvas añade algunas constantes a las que ya tiene la clase Canvas.



**FIRE\_PRESSED:** Representa el bit de la tecla de fuego

**DOWN\_PRESSED:** Representa el bit de la tecla de cursor abajo

**LEFT\_PRESSED:** Representa el bit de la tecla de cursor izquierdo

**RIGHT\_PRESSED:** Representa el bit de la tecla de cursor derecho

**UP\_PRESSED:** Representa el bit de la tecla de cursor arriba

**GAME\_A\_PRESSED:** Representa el bit de la tecla de juego A (si existe)

**GAME\_B\_PRESSED :** Representa el bit de la tecla de juego B (si existe)

**GAME\_C\_PRESSED:** Representa el bit de la tecla de juego C (si existe)

**GAME\_D\_PRESSED:** Representa el bit de la tecla de juego D (si existe)

#### **2.4.2 Layer**

Un layer o capa puede contener una imagen. Además podemos tener más de un layer activo al mismo tiempo, de forma que se superpongan. Por ejemplo, podríamos tener un layer con el fondo del juego (por ejemplo, unas montañas) y otro layer con la imagen del jugador.

Así, podríamos mover la capa del jugador mientras mantenemos estático el fondo dando la impresión de que el objeto de delante se mueve por el escenario.

La clase Layer, es una clase abstracta. Es decir, que no podemos instanciarla directamente.

Hay que crear una subclase. Afortunadamente, MIDP 2.0 nos ofrece dos clases que heredan de Layer y que están listas para usar. Son la clase TiledLayer y Sprite. Por otra parte, necesitaremos una forma de controlar los layers (o sus subclases) para pintarlas en pantalla y poder así superponerlas. Para ello usamos la clase LayerManager.

Los métodos de Layer (y que por lo tanto heredan TiledLayer y Sprite) son:

**int getHeight():** Devuelve la altura del layer

**int getWidth():** Devuelve la anchura del layer

**int getX():** Devuelve la posición x del layer

**int getY() :**Devuelve la posición y del layer

**boolean isVisible():** Si el layer es visible devuelve true.

**void setVisible(boolean visible):** Pone el layer en estado visible o no visible

**void move(int dx, int dy):** Mueve el layer dx pixels en horizontal y dy pixels en vertical respecto a la posición actual

**void setPosition(int x, int y):** Sitúa el layer en la posición x, y

**abstract void paint(Graphics g):** Método abstracto. Debe ser implementado por una subclase.

### 2.4.3 TiledLayer

TiledLayer nos permite crear escenarios con tiles y almacenar todos los tiles en un sólo archivo gráfico (en vez de almacenar los tiles por separado).El archivo gráfico no es importante. Siempre se capturan de izquierda a derecha y de arriba hacia abajo. A cada tile se le asigna un número, empezando por 1.

Algunos métodos interesantes de TiledLayer (además de los que hereda de Layer) son:

**setCell(col, fila, tile):** Establece el tile de una posición concreta de la matriz

**getCell(col, fila):** Devuelve el tile que hay en la posición col, fila

**getCellHeight():** Devuelve el alto del tile

**getCellWidth():** Devuelve el ancho del tile

**getColumns():** Devuelve el número de columnas del TiledLayer

**getRows():** Devuelve el número de filas del TiledLayer

### 2.4.4. LayerManager

La clase LayerManager es la encargada de la gestión de los layers que pertenecen a una misma aplicación. Su misión es mantener una lista de los layers en uso y facilitar métodos para insertar, acceder y borrar layers. En caso de que el layer sea de mayor tamaño que la pantalla,

no permite fijar la posición relativa de la pantalla respecto al layer, pudiendo así crear efectos de scrolling.

Los métodos de LayerManager son:

**append(layer):** Añade un layer al LayerManager.

**getLayerAt(pos):** Devuelve el layer de la posición pos.

**getSize():** Devuelve el número de layers del LayerManager.

**insert(layer, índice):** Inserta un layer en la posición indicada por índice.

**paint(Graphics, x, y):** Dibuja en la pantalla el contenido del LayerManager, en la posición x,y de la pantalla.

**remove(layer):** Borra un layer del LayerManager.

**setViewWindow(x,y,alto,ancho):** Indica la región visible del LayerManager. Es decir, la región que será mostrada al llamar al método paint().

#### 2.4.5 Sprite

Al igual que TiledLayer, Sprite es una subclase de Layer, y la forma en que cargamos los frames que componen la animación del sprite es similar a TiledLayer.

Las almacenamos todas en una imagen que contiene cada uno de los frames.

Al igual que pasaba con TiledLayer, el orden en que se capturan los frames es de izquierda a derecha y de arriba hacia abajo.

En el constructor del sprite indicamos la imagen desde la que vamos a sacar los frames, y el tamaño de cada frame (32x32 píxeles).

Los métodos más interesantes de Sprite (además de los que ya hereda de Layer) son:

**collidesWith():** Chequea la colisión del sprite con algún elemento de la pantalla o con otro sprite. Tiene tres formas:

*-collidesWith(Image image, int x, int y, boolean pixelLevel):* Chequea la colisión del sprite con una imagen, cuya posición x e y debemos facilitarle en el segundo y tercer parámetro. El último parámetro es un valor booleano que indica si el test de colisión ha de hacerse a nivel de pixel (más lento pero más exacto) o mediante su área de colisión.

*-collidesWith(Sprite s, boolean pixelLevel)*: Chequea la colisión con otro sprite.

*-collidesWith(TiledLayer t, boolean pixelLevel)*: Chequea la colisión con un TiledLayer.

Una práctica habitual es crear un TiledLayer para el fondo, y otro con los objetos con los cuales podemos colisionar. Así sólo tenemos que chequear la colisión con éste último.

**defineCollisionRectangle(x, y, alto, ancho)**: Nos permite definir el área de colisión del sprite. Si no definimos nada será todo el tamaño del sprite. Es el área que se comprueba en la colisión si a collidesWith() le pasamos el valor false en el parámetro pixelLevel.

**defineReferencePixel(x, y)**: Con éste método indicamos cual será el pixel de referencia para las operaciones que realicemos con el sprite, como posicionarlo en pantalla, rotación, etc... Si no especificamos ninguno, será la esquina superior izquierda (0,0).

**getFrame()**: Devuelve el índice del frame actual del sprite.

**setFrameSequence(int[] secuencia)**: Establece la secuencia de animación del sprite. Hay que pasar como parámetro un array de enteros con la secuencia de la animación (índice de los frames que la conforman).

Las posibles transformaciones a las que se puede enfrentar un sprite son:

**TRANS\_MIRROR**: Refleja el sprite respecto a su eje vertical.

**TRANS\_MIRROR\_ROT180**: Refleja el sprite respecto a su eje vertical y lo rota 180 grados en el sentido de las agujas del reloj.

**TRANS\_MIRROR\_ROT270**: Refleja el sprite respecto a su eje vertical y lo rota 270 grados en el sentido de las agujas del reloj.

**TRANS\_MIRROR\_ROT90**: Refleja el sprite respecto a su eje vertical y lo rota 90 grados en el sentido de las agujas del reloj.

**TRANS\_NONE**: No realiza ninguna transformación.

**TRANS\_ROT180**: Rota el sprite 180 grados en el sentido de las agujas del reloj.

**TRANS\_ROT270**: Rota el sprite 270 grados en el sentido de las agujas del reloj.

**TRANS\_ROT90**: Rota el sprite 90 grados en el sentido de las agujas del reloj.

### 3. INFORMACIÓN GENERAL DE SOFTWARE EDUCATIVO

#### 3.1 DEFINICIÓN SOFTWARE EDUCATIVO

*Se denomina software educativo al destinado a la enseñanza y el aprendizaje autónomo y que, además, permite el desarrollo de ciertas habilidades cognitivas.* (Brand, R., 1995)

El software educativo, puede y debe estar sujetos a estudio y análisis, ya que su objetivo es apoyar el trabajo pedagógico de profesores y mirar por la captación de nuevos conocimientos por parte de los alumnos. Para asegurarse que el Software Educativo cumpla estos objetivos se debe evaluar sus características en aspectos tan importantes como la calidad de su contenido, la facilidad de uso, la pertinencia de los contenidos, su interactividad, etc. Además, ante el gran incremento de oferta de este tipo de software debemos de intentar desarrollar estrategias de evaluación que hagan eficiente la selección y adquisición de nuestro software para los establecimientos, con el fin de lograr un uso pedagógico significativo.

##### 3.1.1. Tipos de Software Educativo

Existen diversas formas de clasificar software educativo. La clasificación que se presenta a continuación está construida de acuerdo a la forma de utilizar las actividades que presenta cada uno. Al definir los tipos de software que podemos encontrar podemos identificar las diferencias entre cada uno, ayudándonos a decidir para qué tipo de actividad vamos a desarrollar un producto ya que no todos están diseñados con los mismos objetivos. (Pontificia Universidad Católica de Chile, 2000)

TIPO	DEFINICIÓN
EJERCITACIÓN	Programas que intentan reforzar hechos y conocimientos que han sido analizados en una clase expositiva o de laboratorio. Su modalidad es pregunta y respuesta
TUTORIAL	Esencialmente presenta información, que se plasma en un diálogo entre el aprendiz y el computador. Usa un ciclo de presentación de información, respuesta a

	una o más preguntas o solución de un problema. Esto se hace para que la información presentada motive y estimule al alumno a comprometerse en alguna acción relacionada con la información.
SIMULACIÓN	Son modelos de algunos eventos y procesos de la vida real, que proveen al aprendiz de medio ambientes fluidos, creativos y manipulativos. Suele ser utilizadas para examinar sistemas que no pueden ser estudiados a través de experimentación natural.
JUEGO EDUCATIVO	Es muy similar a las simulaciones, la diferencia radica en que incorpora un nuevo componente: la acción de un competidor, el cual puede ser real o Virtual.
MATERIAL DE REFERENCIA MULTIMEDIA	Usualmente presentado como enciclopedias interactivas. La finalidad de estas aplicaciones reside en proporcionar el material de referencia e incluyen tradicionalmente estructura hipermedial con clips de vídeo, sonido, imágenes, etc.
EDUTAINMENT	Tipo de software que integra elementos de educación y entretenimiento, en el cual cada uno de estos elementos juega un rol significativo y en igual proporción. Programas interactivos por excelencia, utilizan colores brillantes, música y efectos de sonido para mantener a los aprendices interesados mientras se les introduce en algún concepto o idea.
HISTORIAS Y CUENTOS	Son aplicaciones que presentan al usuario una historia multimedia, la cual se enriquece con un valor educativo.
EDITORES	El objetivo de estos productos no es dar respuesta

	a preguntas del usuario, sino dar un marco de trabajo donde el alumno pueda crear y experimentar libremente en un dominio gráfico o similar.
HIPERHISTORIA	Software donde a través de una metáfora de navegación espacial se transfiere una narrativa interactiva. Combina un modelo de objetos reactivos en un marco de ambiente virtual navegable. Tiene cierta semejanza con los juegos de aventuras.

*Tabla 1. Tipos de Software Educativo*

El tipo de Software educativo que llevaremos a cabo en nuestra aplicación será el de “Juego Educativo” donde entrará en juego la figura del competidor, que en nuestro caso será un competidor real.

### **3.1.2. Beneficios Software Educativo**

Con un juego de Software Educativo, buscamos el entretenimiento en los más pequeños, ya que, al tratarse de un juego siempre suele divertir a los niños, por otra parte conseguiremos que aumente su habilidad de coordinación viso-manual, puesto que el niño concibe el desarrollo del juego a través de la vista, pero también manualmente a través de sus manos, desarrolla lo que concibe con la vista y esto, se transmite al cerebro. Por otra parte al niño, en este tipo de software continuamente le surgen en la historia del juego una serie de toma decisiones, a través de las cuales hacemos y conseguimos que se incremente la capacidad de respuesta del niño, poniendo a prueba, por otra parte, la capacidad de ingenio y estrategia del niño, viendo como desarrolla lo que llamaríamos su picaresca.(Cabrera, P., 2010).

Estos juegos de ordenador ayudan a enseñar a resolver problemas, establecerse unas metas, y otras habilidades cognitivas importantes. La mayoría de los juegos tienen un aumento de la dificultad con la progresión a los niveles superiores, como suele ser lo normal en la mayoría de los juegos, sea o no lo sean educativos. Esto hace que el niño sienta una necesidad de reto, unas ganas de pasar al siguiente nivel, haciendo que sea más emocionante y le da al niño un sentido de logro con la finalización de cada nivel o fase.

Gracias a la tecnología han surgido muchas maneras innovadoras de conseguir que los niños aprendan de forma diferente a la forma tradicional. Esto provoca en los más pequeños, una nueva forma de entretenimiento, que sin darse cuenta, hace que amplíen y mejoren sus

habilidades lingüísticas, sus habilidades matemáticas y en nuestro caso se les informe y se les enseñe a cómo actuar ante determinados riesgos que les puedan surgir en su vida cotidiana. En general podemos mejorar el conocimiento general de los pequeños a través de estos tipos de Software Educativo.

Se debe tener en cuenta, la edad del niño para seleccionar los juegos educativos más aptos para su edad, así como, qué tipo de cosas o ante que hechos queremos que el niño sepa reaccionar o actuar ante su edad. Esto quiere decir, que a un niño de 5 años le asignaremos un juego educativo que le enseñe los peligros de una casa, por ejemplo, y a un niño de 12 años le asignaremos un juego educativo por ejemplo de educación vial.

### **3.1.3. Modelos de Aprendizaje**

Cada Software Educativo debe estar relacionado con un modelo de aprendizaje, seleccionando aquel que esté más acorde con nuestra aplicación. (Orta,E. et al., 2008)

En nuestro caso llevamos a cabo un tipo de Software de Juego Educativo con un aprendizaje constructivista, es decir desarrollamos un modelo de Aprendizaje denominado “**Construcción**”, debido a que principalmente lo que buscamos es que el niño aprenda y conozca los riesgos naturales y cotidianos que le rodean en su crecimiento.

El modelo de aprendizaje de Construcción es más flexible que los otros, está centrado en el aprendiz y entrega herramientas, materiales, elementos y estrategias para que éste construya y reconstruya su conocimiento. Esto es principalmente sustentado por el hecho que el aprendiz, para trabajar con el software, debe hacer cosas, construir, reconstruir, resolver, crear, corregir y reparar los errores. El aprendiz hace cosas con el software y no el software hace cosas con él. En este tipo de software, además de considerar un tratamiento flexible y dinámico, existe una intencionalidad de desarrollar o estimular el uso de algún proceso cognitivo y su transferencia al aprender. Aquí el aprendiz juega, se entretiene, resuelve complejidades, controla variables, se enfrenta a situaciones inciertas, resuelve problemas, etc. Algunas ilustraciones de este tipo de software son juegos educativos y de entretenimiento (edutainment), cuentos e historias hipertextuales e interactivas, simulaciones, editores, etc. En todos ellos, se incorporan estrategias cognitivas que motivan e involucran al aprendiz a través de otorgar control sobre la tarea de aprendizaje, desafío, interacción y adaptación al nivel y requerimiento del aprendiz. En general, los distintos tipos de software responden a un modelo de aprender.

Los software de presentación muestran un modelo de estímulo - respuesta, los de representación un modelo de estructuración en memoria semántica o conceptual y los de construcción un modelo activo de aprender y conocer. Debemos tener en cuenta estos modelos



para realizar una evaluación ajustada al tipo de software revisado, al tipo de metodología y a las actividades que pretendemos trabajar con los alumnos.

#### 3.1.4. Evaluación de Software Educativo

La evaluación del software educativo se debe enfocar de tres formas. La primera considerando los atributos genéricos que se deben tener siempre en cuenta a la hora de evaluar, una segunda consiste en tener en cuenta el punto de vista del usuario y por último debemos tener en cuenta aspectos concretos propios de una evaluación de Software.

A la hora de llevar a cabo el análisis de un Software Educativo es importante considerar los siguientes atributos (Enlaces, Centro de Educación y Tecnología Chile)

**Constructividad.** Se refiere a la posibilidad de construir nuevos escenarios a partir de la combinación de objetos en espacio y tiempo.

El aprendiz hace cosas, construye, tiene actividad. El desarrollo del software depende de las acciones que el aprendiz haga, de las decisiones que tome. Este concepto, está muy ligado a la teoría de aprendizaje constructivista.

**Navegabilidad.** Posibilidad de explorar libremente los ambientes que componen el mundo, dominio o estructura de información presentada.

**Interactividad.** Se define como la capacidad dinámica que refleja un sistema, el cual:

- \* Provee retroalimentación al usuario en tiempo real
- \* Adapta o modifica dinámicamente su comportamiento en función de los eventos recibidos
- \* Entabla alguna modalidad conversacional con un cierto grado de detalle, complejidad y modalidad.

La interactividad de un dispositivo, sistema o medio en particular, puede ser medida por *“la cantidad de información intercambiada, por la retroalimentación (feedback) objeto y sujeto, por la acción y reacción. La interactividad también está definida por el grado de manipulación y control posible sobre las variables del mundo y, por el nivel de respuesta percibida por el usuario”*. Indudablemente, una mayor interactividad está dada por un software que habilita una mayor actividad por parte del usuario, esto es, un software que otorgue mayor control, mayor cantidad de herramientas para hacer y construir cosas, esto es un software de tipo construcción.

**Contenido.** Es la calidad, fiabilidad, organización y relevancia de la información entregada. Es un atributo ortogonal a la presentación y, dependiendo de la audiencia, debe ser adaptado y organizado.

**Interfaz.** Es la superficie de contacto entre el aprendiz y computador. Es la pantalla con la cual el aprendiz interactúa. Es el modo de capturar la acción y atención del aprendiz y de reflejar el estado y contenido del sistema.

La interfaz tiene fuerte impacto en la navegación, construcción e interactividad provista.

Desde el punto de vista del usuario (aprendiz), los descriptores más relevantes en un software educativo se relacionan con la capacidad de:

\*Desafiar al aprendiz

\*Interactuar con él o ella

\*Motivarlo

\*Jugar con él o ella

\*Envolverlo

\*Entretenerlo

\*Estimularlo para realizar asociaciones entre información de distinto tipo

\*Adaptarse a los niveles y ritmos del aprendiz

\*Darle el control de los eventos del software y posibilidades de evaluación permanente

\*Darle herramientas y materiales para diseñar, crear y construir

\*Estimular en él o ella el descubrimiento

\*Estimular el desarrollo de la imaginación, la capacidad de maravillarse y de construir fantasías

Por otra parte, deberemos de analizar los siguientes criterios para saber si la utilidad del software nos va a aportar lo que buscamos en él.

**Pertinencia.** Se encuentra estrechamente relacionada con la coherencia que existe entre el contenido del software y aspectos como:

\*Modelo de aprendiz

\*Políticas educativas del país

\*Modelos curriculares de cada entidad educativa

\*Contenidos y destrezas que se espera usar y desarrollar

\*Concepciones constructivistas del aprendizaje

**Orientaciones Metodológicas.** Tiene relación con el material de apoyo educativo adicional que acompaña al software para orientar tanto al aprendiz como al educador.

**Utilidad.** Está relacionado con lo eficiente que puede ser el software en función a otros medios pedagógicos utilizados en forma tradicional y a lo funcional que puede ser esta herramienta para el grupo de usuarios finales.

**Adaptabilidad.** Tiene relación con las posibilidades de adaptar el contenido del software a una diversidad de usuarios, sus niveles y al desarrollo de objetivos diferentes a los planteados originalmente por éste.

**Usabilidad.** Está relacionado con el nivel de complejidad y el agrado que nos produce el programa, las posibilidades de modificación, la forma como lo calificamos, la facilidad de uso, la facilidad de recuerdo, los errores que existan y las recomendaciones que daríamos para su uso.

Como vemos, esto dependerá de la motivación que nos produzca, la cual está vinculada a la complejidad del manejo del software y el nivel de atención que se requiere por parte del usuario.

Debido a la importancia de este punto, posteriormente lo desarrollaremos más específicamente indicando los resultados propios de la usabilidad de nuestro videojuego educativo

**Interactividad.** Se relaciona con el nivel de comunicabilidad que tiene el programa y los diálogos que se establecen entre éste y el usuario. Esto dependerá en gran medida de las pantallas del programa y los medios que utiliza (imágenes, textos, sonido y video).

**Modalidad.** Tiene relación con la forma en que el aprendizaje es enfocado por el programa, es decir, si entrega contenido, si permite la ejercitación, si simula procesos, si estimula aprendizajes, etc.

## 4. USABILIDAD

---

### 4.1. INTRODUCCIÓN

Norman & Drapper, 1986; Norman, 1988 (Zurita, G., 1999, Sánchez, J., 1999, Nussbaum, M., 1999) *concuerdan que, durante y posterior al ciclo de diseño y desarrollo de software, es saludable y necesario realizar algún tipo de evaluación con el usuario final.*

La usabilidad intenta responder a la pregunta de cómo los usuarios pueden usar adecuadamente la funcionalidad de una interfaz de software, es decir, cuán bien los usuarios responden a los aspectos que el sistema le ofrece.

Por otra parte, tanto los niños como los adultos tienden a buscar situaciones agradables al interactuar con un juego, por lo que su naturaleza lúdica debiera considerar un diseño entendible y que genere un modelo mental directo en el niño.

Es importante identificar y analizar las posibilidades de mejoría de la estructura del diseño visual y funcional de los juegos educativos, para facilitar de mejor forma el aprendizaje implícito de su contenido. Ruiz-Vargas, 1991(Zurita, G., 1999, Sánchez, J., 1999, Nussbaum, M.,1999) sostienen que *aquellos aspectos con mayor contenido visual son mejor recordados sostienen además, que se recuerdan mejor aquellos aspectos que son agradables, es por ello que las pantallas con las que el usuario interactúa, presentan dibujos, los cuales muestran pocos detalles y colores contrastados para que el niño los pueda asimilar, así como, las pantallas donde el usuario obtiene información de cómo jugar o de la situación que se le presenta en el juego, todas ellas, presentan el mismo formato diferenciándose tan solo en el color del fondo de la pantalla.*

Teniendo en cuenta que Ruiz-Vargas sostienen que *en general la inmadurez tiene como consecuencia un déficit atencional y si no existe atención se adquiere menos información y por lo tanto se recuerda menos, como consecuencia de ello se aprende menos, luego, un estudio de usabilidad de la interfaz aplicado a niños, identificará aquellos aspectos que permitan hacer que estos presten mayor atención y conseguir mejores resultados con el uso del juego educativo*

### 4.2. MÉTODOS

El juego se ha desarrollado en base a que predisponga de una buena usabilidad, para ello se toman determinadas medidas.

Entre estas medidas, se destaca el mostrar dibujos simples, para facilitar la comprensión al niño, también se usa un contraste entre el color de fondo de pantalla y el de las letras

Los métodos elegidos para medir la usabilidad del videojuego educativo “Un día con Tico” han sido la observación y la entrevista.

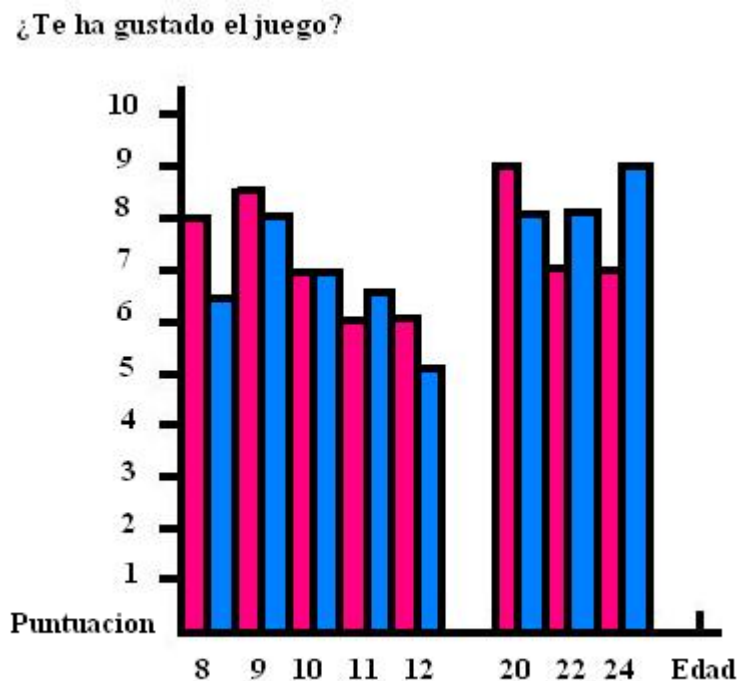
Se usó el método observacional para captar la destreza de los niños a la hora del desarrollo del juego, ayudándose principalmente a adecuar el juego a las características propias de los niños a los que va destinado y siendo consciente de la diferencia en el desarrollo que puede existir entre un niño de 8 años con uno de 12 años.

La entrevista fue realizada posteriormente a la parte práctica, de una forma abierta, en la cual se ha obtenido una gran satisfacción porque realmente se visualizaron determinados aspectos que no se habían abarcado en el juego hasta que un determinado niño lo ha contado.

La entrevista oral me ha demostrado ser muy adecuada para poder interactuar con niños, en comparación a métodos de registro escrito como los cuestionarios. Los niños muestran emociones, comentarios y conductas complejas de registrar mediante métodos más cuantitativos.

La muestra estuvo formada por diez niños con edades comprendidas entre los 8 años y los 12 años. Esta muestra esta diferenciada por sexo y por edad y además otra parte de la muestra esta formada por un total de 5 adolescentes con una edad mayor de 18 años y menor de 24 años pero que presentan una disfunción sensorial.

#### 4.3. RESULTADOS



*Imagen 3. Resultados 1*

Por lo general, tal y como refleja la estadística el juego les ha gustado a ambos grupos de personas encuestadas (*niños de 8 a 12 años y adolescentes con disfunción sensorial de 20 a 24 años*), queriendo indicar que existió una mayor expresividad en los niños de menor edad y en los chicos con problemas sensoriales.

También indicar que en contactos previos con ciertos niños, mostraron ciertos comentarios que me ayudaron a crear diferentes pantallas del juego y cuyas propuestas como la pantalla de los cristales en el suelo, fueron realmente pensadas y propuestas por ellos.

En los chicos de mayor edad, es decir con 11 y 12 años se notó en ellos que mostraban atención al juego pero que realmente el juego es un poco infantil para su edad, debido a que están acostumbrados a desarrollar con esa edad otro tipo de juegos y aplicaciones en sus móviles.

Nivel de dificultad (1: fácil - 10: difícil)

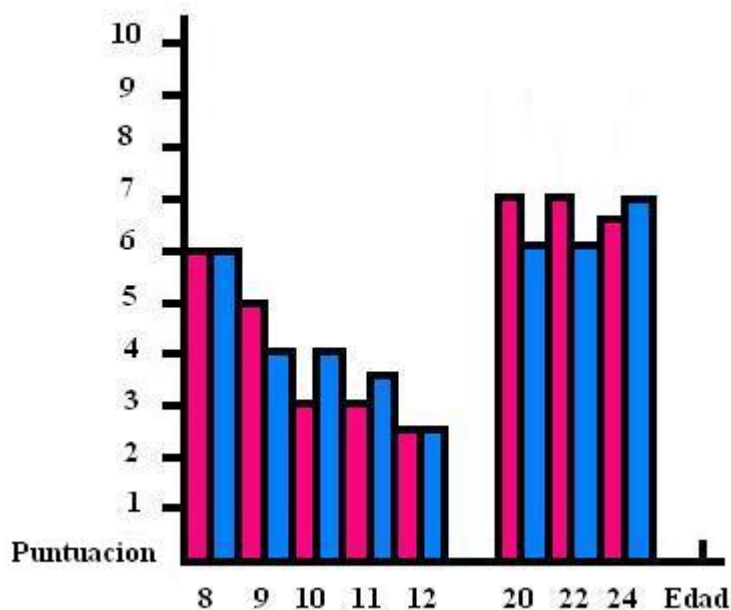


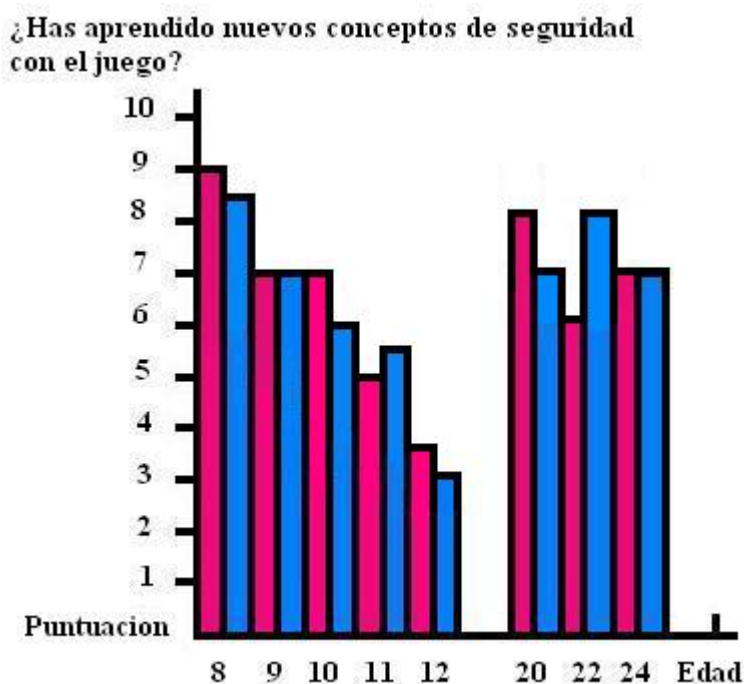
Imagen 4. Resultados 2

Como vemos en la gráfica, se nota cierta diferencia a la hora de asimilar o de reflejar la dificultad del juego entre el grupo de niños y el grupo de chicos con problemas sensoriales.

Mientras que el grupo de chicos con problemas sensoriales asimilan la dificultad más o menos con el mismo grado, debido a que dentro de sus posibilidades todos intentaron desarrollar el juego de inicio a fin y digamos que su destreza con la manejabilidad que propone el móvil es similar, en el grupo de los niños, vemos como a los más pequeños les resultó más difícil las diferentes pruebas que propone el juego que a los más mayores.

Destacar que a dentro del grupo de los niños, en todas las edades, lo que mayor dificultad les presentó fue la parte del bonus, puesto que digamos, presenta alguna prueba que es menos común de conocimiento o cotidiana de conocer.

Otra cosa que me gustaría destacar, es la diferencia de velocidad en la lectura entre ambos grupos y dentro del grupo de los niños, reflejar que existe una gran diferencia entre la lectura desarrollada por un niño de 8 años y otro de 12 años, de ahí, que el paso entre pantalla y pantalla se realice a través del número 0, puesto que estableciendo un tiempo de visualización de pantalla, en pruebas previas me hizo ver que no todos los niños eran capaces de leer en tres segundos lo que informaba la pantalla.



*Imagen 5. Resultados 3*

En este caso, aunque la estadística refleje varias notas elevadas, sobre todo entre los más pequeños y el grupo de chicos con problemas sensoriales, en mi opinión, creo que realmente los datos de los grupos de 10, 11 y 12 años reflejan la realidad.

Con esta opinión personal, no quiero reflejar que el juego no eduque o no enseñe, porque por lo contrario pocos fueron los que consiguieron la máxima puntuación, pero sí que hay puntuaciones como en las niñas de 8 años en donde se obtiene una media de 9 puntos, que a mi parecer me parece excesiva, pero pienso que también esa puntuación se debe a que la última prueba del juego, en la cual, la cocina empieza a arder y te pide que indiques como huir, es una prueba la cual muchos niños no han sabido contestar y es como que al ser la última prueba difícil les influyo en su respuesta posteriores de la encuesta, pienso yo.



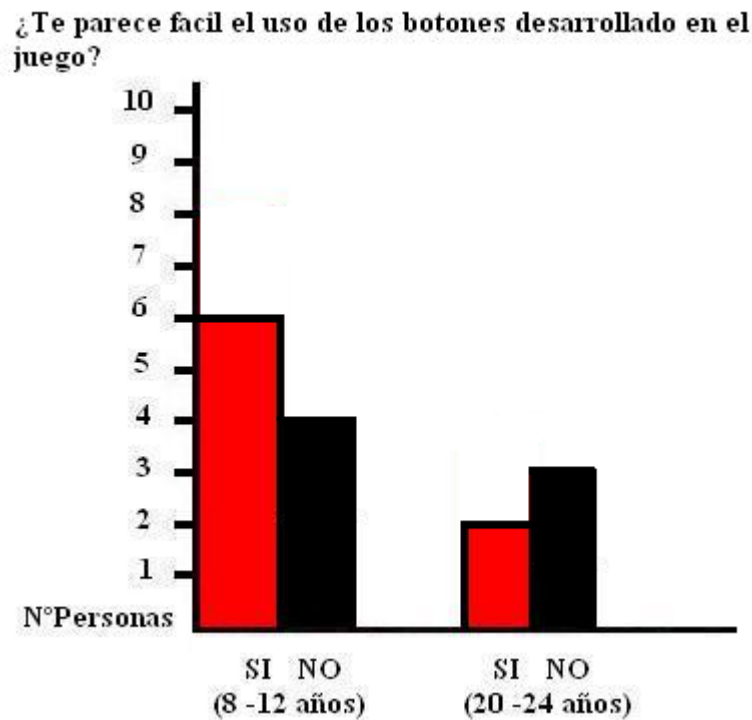


*Imagen 6. Resultados 4*

En esta ocasión, e indiferentemente de las respuestas, personalmente pienso que no he sabido hacerlo con mayor y mejor usabilidad, es decir, creo que el juego está desarrollado de una forma perfecta para que cualquier niño con problemas auditivos o visuales pueda desarrollar una partida en perfectas condiciones.

Los niños que me respondieron que NO, hicieron hincapié en el tamaño de la letra, lo cual, lo respeto pero al no existir scroll en la pantalla para hacer más fácil su manejo, el aumento del tamaño de la letra me era imposible.

También se comentó el retraso que se produce en el juego, cuando el sonido está encendido a lo cual, dentro de un lenguaje asequible para ellos, les hice entender que debido a ciertos problemas a la hora de desarrollar el juego no pude conseguir evitar esos retardos.



*Imagen 7. Resultados 5*

Tras estos resultados, puede verse que en ambos grupos de encuestados, se difiere en las respuestas y no existe tanta diferencia entre el SI y el No.

A lo largo del juego e intentado usar las mismas teclas y siempre usando una línea del teclado del móvil o en cruz, pero el existir diferentes número de teclas en las pantallas se hace imposible el usar siempre las mismas teclas, de ahí que en algunas pantallas se use 3 teclas y en otras 4 principalmente.

Por lo general, tras verles jugar, tampoco he notado una dificultad que no vaya más allá de mirar el teclado para ver dónde está y dónde se encuentra el número que quieren pulsar.

## 5. ACCESIBILIDAD

---

Siempre deberemos tener en mente los problemas de accesibilidad en el diseño de cualquier tecnología orientada a la población para evitar discriminaciones, debido a que hay que considerar a los muchos usuarios que por diversos motivos y disfunciones fisiológicas o sensoriales, no puedan operar en un contexto normal y se les deba de facilitar la accesibilidad para que también ellos puedan actuar sobre estos contextos:

- Pueden no ser capaces de ver, escuchar, moverse o pueden no ser capaces de procesar cualquier tipo de información.
- Pueden tener dificultad en la lectura o comprensión de un texto.
- Puede ser que no puedan ser capaces de usar un teclado o un ratón

Debido a este problema y a que contamos con una sociedad en constante evolución para responder a estas necesidades en los últimos años a aumentado el interés en hacer y conseguir la mayor accesibilidad en estas nuevas tecnologías y como no, también en el ámbito del software educativo.(Sanz-Troyano, E., et al, Moreno-Ger, P., et al)*Debemos saber, no obstante, que la introducción de estas nuevas tecnologías hacen surgir nuevos problemas de accesibilidad pudiendo poner en riesgo el derecho universal a la educación para alumnos con discapacidad.* Dejando a un lado este problema y en cuanto a los videojuegos nos atañe, debemos saber que tanto los videojuegos como las herramientas de autoría de videojuegos han sido propuestos como medio para fomentar el desarrollo de nuevas habilidades, indicándonos con esto, que son válidos para fomentar la enseñanza entre personas con accesibilidad reducida.

En campos tan importantes como la educación donde también se han incorporado tecnologías integradas con softwares educativos y donde se han desarrollado estrategias pedagógicas con estos nuevos softwares se debe tener especial cuidado de no discriminar a estas personas con dificultades de accesibilidad para que ellos también puedan adquirir conocimientos con este nuevo tipo de tecnología. En la actualidad sabemos que se intenta proporcionar para los estudiantes videojuegos educativos dado su potencial como herramienta de aprendizaje, usándolos por ejemplo, para mejorar el aprendizaje mediante la interacción del alumno con los videojuegos o como elemento de aprendizaje mediante la propia creación de videojuegos, como es el caso.

## 5.1 ACCESIBILIDAD EN LOS JUEGOS

En el caso de la accesibilidad Web, existen iniciativas como son WAI<sup>1</sup> o WCAG, el problema lo encontramos para otro tipo de tecnologías, no basadas en la Web, debido a que no existen pautas ampliamente aceptadas sobre accesibilidad en otros campos, como puede ser los videojuegos.

Existe también otra iniciativa a destacar en nuestro ámbito pero es menos conocida que las anteriores, es la ATAG o Authoring Tool Accessibility Guidelines, que se centra en estudiar cómo hacer accesibles herramientas de autoría para diseñar contenido Web accesible. Principalmente es una iniciativa preocupada por la accesibilidad del campo web también pero pueden tomar sus premisas los desarrolladores de herramientas de autoría para obtener interfaces de autoría accesibles.

En general, el tema de la accesibilidad es uno de los asuntos menos abordados en la industria de los videojuegos, ya sea por una falta de concienciación, o porque a la industria del videojuego no le rentabiliza la inversión en materia de accesibilidad, el resultado es que los videojuegos son poco accesibles para personas con discapacidad.

No obstante, también existen escasas iniciativas hardware y software que tratan de mejorar la accesibilidad de los videojuegos.

Centrándonos en el hardware, la accesibilidad se intenta conseguir incorporando dispositivos de soporte como son mandos de juego adaptados para personas con distintos tipos de discapacidad, o incluso sistemas que reconozcan el movimiento de los ojos o de la cabeza. También existen teclados especiales que permiten de forma alternativa generar comandos de teclado a través de un sistema hardware o software. Estos teclados pueden tener una separación de teclas muy grande, teclados que se visualizan en la pantalla o incluso teclados activados por el movimiento del ojo.

En la actualidad, la accesibilidad mediante dispositivos externos es la forma más común de conseguir una buena accesibilidad, debido en parte, a que incluir nuevas funcionalidades de software requiere un coste añadido que la industria no suele estar dispuesta a asumir ya que este sobre coste se produciría en cada nuevo videojuego.

Si este problema lo tratamos en el caso de los juegos educativos, vemos que el problema en sí se agrava, debido a que los presupuestos son aún más limitados y encima la importancia de

---

<sup>1</sup> <http://www.w3.org/WAI/>  
<http://www.w3.org/TR/WCAG10/>

la accesibilidad es aún mayor. Una alternativa muy utilizada para solucionar este sobre coste en los juegos, es utilizar herramientas (también software) de asistencia, como lectores de pantalla, magnificadores, sintetizadores/reconocedores de voz (en el caso de presentar discapacidad motora) o teclados virtuales. Otra alternativa utilizada consiste en diseñar la interacción con el videojuego de forma multimodal, es decir combinar, por ejemplo, subtítulos e imágenes con sonidos y voces, interactuando con el videojuego a través de comandos de voz. Pero este tipo de soluciones suelen llevarse a cabo de forma *ad-hoc*, por lo que conlleva igualmente un aumento del coste de producción del juego debido por ejemplo a la sonorización de los diálogos entre personajes, lo cual vuelve a llevarnos al gran problema de la limitación del presupuesto.

A continuación se muestra las diferentes deficiencias funcionales, donde se nos indica las posibles soluciones de accesibilidad que se proponen para cada caso y de esta forma facilitar al usuario la intermediación con el juego. (Mairena, J., 2009)

### **5.1.1 Diversidad Funcional Auditiva**

La diversidad funcional auditiva proporciona múltiples soluciones para dar accesibilidad en los videojuegos debido a que en la actualidad la mayoría de los videojuegos son principalmente visuales.

#### *Características de la diversidad funcional auditiva*

La diversidad funcional auditiva consiste en la pérdida parcial o total de la audición, pudiendo existir diversos grados de nivel de pérdida de audición.

Cuando el nivel de pérdida de audición es elevado, resulta complicado seguir una conversación debido al ruido de los agentes externos que nos rodean.

Si el nivel de pérdida de audición es intermedio, nos podemos servir de algún tipo de ayuda auditiva, como un audífono, y en los más graves, donde la pérdida de audición es más elevada, además de una ayuda auditiva, es necesario leer los labios o usar el lenguaje de signos.

#### *Técnicas de accesibilidad para la diversidad funcional auditiva*

- \* *Subtitular todos los diálogos*
- \* *Closed Caption [CC]:* Consiste en subtitular todos los sonidos que se escuchan en el juego, no sólo los diálogos.
- \* *Volumen configurable*

### **5.1.2 Diversidad funcional cognitiva**

#### *Características de la diversidad funcional cognitiva*

Dentro de los diferentes tipos que nos podemos encontrar en la diversidad funcional cognitiva destacamos los siguientes:

- \* *Pérdida de memoria*
- \* *Déficit de atención*
- \* *Dislexia*
- \* *Trastornos Generalizados del Desarrollo*

#### *Técnicas de accesibilidad para la diversidad funcional cognitiva*

- \* *Velocidad configurable*
- \* *Varios niveles de dificultad*
- \* *Grandes iconos o fotos para los menús de navegación*
- \* *Indicadores hacia donde ir*
- \* *Niveles de entrenamiento*
- \* *Textos hablados con velocidad ajustable*
- \* *Usar un lenguaje y vocabulario sencillo*
- \* *Opción de pausa en cualquier momento*

### **5.1.3 Diversidad Funcional Visual**

#### *Características de la diversidad funcional visual*

Podemos dividir la diversidad funcional visual en tres tipos diferenciándolos en:

*Ceguera:* Es la imposibilidad de ver. Estos jugadores necesitan comunicarse con el juego a través de medios no visuales, como puede ser el sonido o la vibración de un mando.

*Baja visión:* Muchas personas con diversidad funcional visual conservan algo de visión y pueden usar un ordenador magnificando partes de la pantalla y destacando lo importante que queramos representar en el videojuego.

*Ceguera de colores:* La ceguera de colores consiste en la imposibilidad y dificultad que tiene una persona para diferenciar ciertos colores.

En contra de lo que pensamos una gran parte de la población, no sólo existen los casos en los que se confunden el rojo y el verde, o el azul y el amarillo; hay muchas personas que estos colores los diferencian pero no puede diferenciar ciertos colores como, por ejemplo, algunos tonos de verdes oscuros de otros tonos de marrones oscuros o grises.

También hay personas que sólo ven tonos de grises.

#### *Técnicas de accesibilidad para la diversidad funcional visual*

*Textos claros:* Deberemos de aplicar un tipo de letra sencillo y de un tamaño adecuado para facilitar la lectura.

*Voces en varios idiomas:* Esta opción se podría aplicar con el fin de que cada jugador pueda oír los diálogos en su idioma y no lo limitemos simplemente a leer los textos y oírlos en castellano.

*Modo de alto contraste:* La definición gráfica del juego deberá ser alta para que los jugadores con problemas en la visión puedan diferenciar con mayor claridad cada elemento del juego.

*Modo de gráficos simples:* Haremos que los gráficos empleados sean mucho más sencillos que los originales, sustituyendo las figuras geométricas por otras que sean más simples.

También existe la posibilidad de eliminar completamente los fondos de la pantalla para evitar posibles confusiones.

*Posibilidad de aumentar los gráficos:* Debemos de estudiar si existe la posibilidad de cambiar el tamaño de los elementos gráficos del juego, aumentándolos, pero siempre teniendo cuidado de que la forma de jugar va a seguir siendo sencilla.

*Sonidos indicadores:* Intentaremos conseguir que la mayor parte del juego pueda efectuarse a través de sonidos y que se pueda jugar con sólo oírlo, esto no quiere decir que sea

bueno hacer juegos que ni siquiera tengan gráficos, como muchos de los que se pueden encontrar en Internet pensados para esta diversidad funcional.

Sin gráficos, debemos saber que el juego no resultaría atractivo para el resto de jugadores y el videojuego podría perder su sentido socializador.

#### **5.1.4 Diversidad Funcional en la Movilidad**

##### *Características de la diversidad funcional en la movilidad*

Dentro de la diversidad funcional en la movilidad se pueden encontrar tres características que pueden dificultar la accesibilidad a los juegos actuales:

- \* *Pérdida de movilidad*
- \* *Movimientos repetidos*
- \* *Pérdida de velocidad, fuerza o precisión*

##### *Técnicas de accesibilidad para la diversidad funcional en la movilidad*

- \* *Control con sólo un botón*
- \* *Control con una sola mano*
- \* *Evitar la pulsación de varios botones simultáneos*
- \* *Sensibilidad de los controles configurable*
- \* *Velocidad configurable*
- \* *Varios niveles de dificultad*
- \* *Control por micrófono*
- \* *Usar periféricos baratos y ya extendidos*



## **6. RIESGOS INFANTILES: INFORMACIÓN Y ACTUACIÓN**

---

El grupo de niños con edades comprendidas entre los 3 y 5 años son los más expuestos a los riesgos infantiles. Éste no depende de si mismo, debido a su corta edad y no sabe detectar ni darse cuenta de la existencia de un peligro. Principalmente a causa de este motivo la prevención debe desarrollarse más. Por este motivo, la acción preventiva debe desarrollarse con más atención por parte de todas las personas implicadas en la acción educativa.

Tras el paso de los años los cambios que se realizan en diferentes aspectos de la vida cotidiana para mejorar nuestro nivel de vida aumentan, pero muchas veces aparecen factores de riesgo producidos por estos cambios, sin tener en cuenta las necesidades infantiles y sin pensar en los posibles riesgos infantiles que se generan por estas mejoras.

Factores como la mejora y aumento de la movilidad y el transporte, la mecanización doméstica, la electricidad, el tipo de vivienda, diseño inmobiliario, entornos domésticos y presencia en el hogar de sustancias tóxicas han conseguido facilitarnos la vida en general, pero no han tenido en cuenta la presencia de los pequeños en el hogar.

## 7. HISTORIA DEL JUEGO “UN DÍA CON TICO”

---

### 7.1 PRESENTACIÓN

Teniendo en cuenta los factores anteriormente mencionados y la oportunidad de desarrollar una actividad lúdica, desarrollamos este juego educativo con el objetivo principal de conseguir que los más pequeños consigan asimilar determinados riesgos que se pueden encontrar en su vida cotidiana y como actuar ante estos riesgos y ante los posibles riesgos naturales que nos pueden surgir en nuestra vida.

Conseguir que un niño determine, por ejemplo, que la lejía es un producto peligroso o que, después de hacer unas tostadas debemos ser conscientes de que hay que apagar el fuego de la cocina, sería lo más importante y más bonito que se pueda esperar de este juego educativo.

“UN DÍA CON TICO” es el nombre elegido para nuestro juego digital, a través del cual queremos enseñar a los más pequeños las formas de actuación y prevención ante determinados riesgos que surgen en el hogar que nos pueden surgir con mayor o menor frecuencia.

Tico será un perro al cual le irán surgiendo diferentes acciones, diferentes situaciones y dará la posibilidad al joven usuario de decidir la mejor forma de actuar ante estas situaciones intentando ir salvando todos estos obstáculos para llegar al fin del juego sano y salvo.

La historia del juego se narrará comenzando con el inicio de un día, desde la mañana, hasta la hora de acostarse, siendo este acto, el que nos indique el fin y consecución de los objetivos que nos marca el juego.

Durante el transcurso de la partida, Tico, realizará acciones y le irán surgiendo situaciones a las cuales se puede enfrentar un pequeño cada día de su vida, pero, el elegir un perro como protagonista de la historia, se debe a que, utilizaremos a esta mascota, como instrumento para poder mostrar a los niños a través del juego como resolver cada situación, puesto que habrá situaciones que un niño de corta edad no debe o no sabrá actuar, pero usando una mascota como actor del juego, podemos hacer que resuelva estas situaciones y todas las que se nos ocurran, por eso es un personaje irreal.

Por otra parte, indicamos, que la elección de un perro como mascota, se debe también a la cercanía de muchos niños hacia este animal, puesto que es el animal más común en nuestros hogares y todos sabemos que suelen ser animales muy cariñosos y dóciles con los más pequeños de la casa.

## 7.2 DESARROLLO DEL JUEGO

El juego se desarrolla en diferentes pantallas, donde cada una muestra una situación inicial que le surge a nuestro personaje, y en donde el usuario debe interactuar para solventar la situación de la mejor forma.

A continuación explicaremos cada pantalla en las que se divide nuestra aplicación

*PANTALLA INICIAL:* Antes de comenzar el juego mostramos una pantalla inicial, dando la posibilidad al usuario de apagar o encender el sonido, visualizar ranking de puntuaciones, salir de la aplicación, acceder a la ayuda o comenzar una nueva partida.

Si el usuario elige una nueva partida, aparece una nueva pantalla contando una breve historia sobre nuestro personaje Tico, muestra a continuación la primera pantalla del juego en la que el usuario interactuará.



*Imagen 8. Pantalla Inicial*

*EL DESPERTAR:* Tico despierta después de pasar una plácida noche durmiendo en su cama para desayunar.

Al despertarse deberá recoger su habitación, indicando el usuario que cosas debe recoger y cuáles no.

La ventana de la habitación está abierta, el usuario deberá cerrarla para prevenir que un soplo de aire pueda romper algo en la habitación.

La calefacción debemos de apagarla, puesto que cerca hay ropa y no podremos salir de la habitación dejando la calefacción encendida porque se puede prender la ropa.



*Imagen 9. El Despertar*

**EL DESAYUNO:** En esta pantalla, el usuario debe indicar a Tico que dos alimentos son los más adecuados para llevar una alimentación sana, evitando aquellos alimentos que contienen grasas industriales o refrescos con gas.



*Imagen 10. El Desayuno*

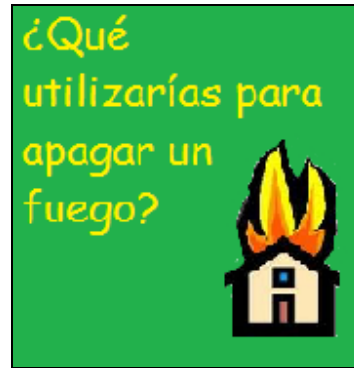
**BONUS:** Hemos acabado de desayunar y ahora vamos a dar la oportunidad al usuario de obtener unos puntos extras haciéndole unas preguntas, cuyas respuestas son muy valiosas para poder saber cómo actuar ante una situación de emergencia.

En la primera imagen de cada Bonus, se muestra durante 4 segundos unos dibujos enumerados que dan contestación a la pregunta que se le formula después al usuario, obligando al usuario a memorizar los dibujos para poder responder a la pregunta.

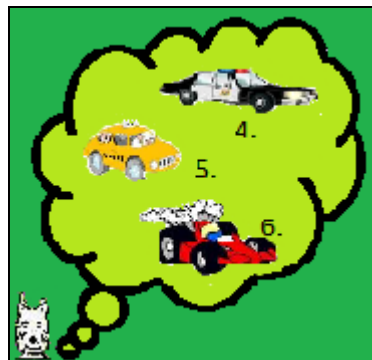
En esta pantalla, no se le da la oportunidad al usuario de poder volver a repetir el Bonus, puesto que debido a la alta puntuación que se obtiene de él, a la hora del ranking el mostrar interés en esta pantalla es de gran importancia para desmarcarse en la clasificación. Además al finalizar el Bonus, se muestra las respuestas correctas e incorrectas del Bonus, para que el usuario pueda asimilar los resultados obtenidos.



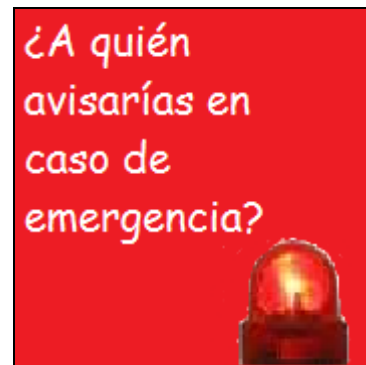
*Imagen 11a. Bonus I Dibujo*



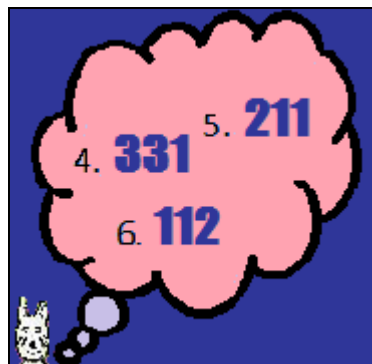
*Imagen 11b .Bonus I Pregunta*



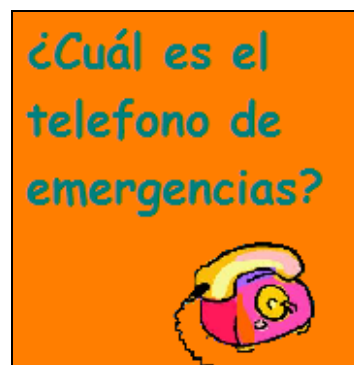
*Imagen 12a. Bonus II Dibujo*



*Imagen 12b .Bonus II Pregunta*



*Imagen 13a. Bonus III Dibujo*



*Imagen 13b .Bonus III Pregunta*



*Imagen 14a. Bonus IV Dibujo*



*Imagen 14b. Bonus IV Pregunta*

**PREPARAR EL BAÑO:** Tico se va a duchar, pero antes debe preparar el baño, y el usuario debe elegir entre varios productos que aparecen en la estantería. Aparecen productos tóxicos, y no tóxicos como el gel y el champú. Si el usuario ha seleccionado los productos tóxicos y que no pueden estar en el cuarto de baño, verá conseguida una bonificación de puntos.



*Imagen 15. Preparamos el Baño*

**TICO SE DUCHA:** Es hora de darse una ducha, pero la pantalla nos visualiza una situación que no nos permite ducharnos sin el temor de que pueda dar una descarga eléctrica mientras Tico toma su ducha. El usuario deberá de indicar que acción tomar para poder seguir sumando puntos a su casillero.



*Imagen 16 .La Ducha*

*CAMINO A LA COCINA:* Tico ha acabado su ducha y ahora le apetece ir a la cocina para tomar algo de media mañana, pero por el camino se encuentra con la ventana del pasillo rota. Al usuario se le propone que nos indique la forma más adecuada para que Tico pueda llegar a la cocina y tomar su almuerzo de media mañana sin cortarse por el camino



*Imagen 17. Cristales Rotos*

*INCENDIO COCINA:* Finalmente, Tico ha conseguido recoger los cristales y llega a la cocina para tomar su almuerzo, pero de nuevo un incidente hace que no pueda almorzar. Tico en esta ocasión no podrá por si solo arreglar el incidente, pero el usuario si le deberá indicar la forma más correcta de huir de tan problemática situación.



*Imagen 18. Incendio Cocina*

**ANIMAL PUNTUACIÓN:** Hemos acabado el juego, ahora el usuario debe elegir el animal con el cual desea grabar su puntuación. La forma de grabar la puntuación a través de un animal, la realizo de esta forma, para poder dar la posibilidad al usuario con deficiencias auditivas de saber con qué datos guarda su puntuación y así a la hora de escuchar el ranking pueda diferenciar cual es su animal con sus puntos correspondientes.



*Imagen 19. Elección animal Ranking*

**INFORMACIÓN PUNTOS:** Una vez elegido el animal y haber finalizado por completo la partida del juego, se nos visualiza una pantalla final recordando el animal elegido para guardar nuestra puntuación y además nos informa de los puntos que hemos obtenido a lo largo de la partida.

Después de esta pantalla, el juego vuelve a su menú inicial pudiendo el usuario realizar la acción que desee.





Imagen 20.Datos animal y puntos

**RANKING:** A esta pantalla el usuario podrá acceder desde el menú principal y en ella verá reflejada las últimas 5 puntuaciones más altas que se hayan producido a lo largo de las partidas que se hayan desarrollado en el juego.

Una pantalla con fondo rojo. En la parte superior, el título "RANKING" está subrayado. Debajo, hay una tabla con dos columnas: "NOMBRE" y "PUNTOS".

	<u>NOMBRE</u>	<u>PUNTOS</u>
1	Elefante	120
2	Tico	100
3	Gato	80
4	Cerdo	60

En la parte inferior izquierda hay un pequeño icono de un perro blanco, y a su derecha hay cinco huellas de pata blancas apiladas horizontalmente.

Imagen 21 .Ranking

**AYUDA:** A esta página de ayuda, el usuario puede acceder desde el menú de inicio del juego y en ella puede informarse sobre la manejabilidad y uso de los botones para poder llevar a cabo una partida mejor informada.

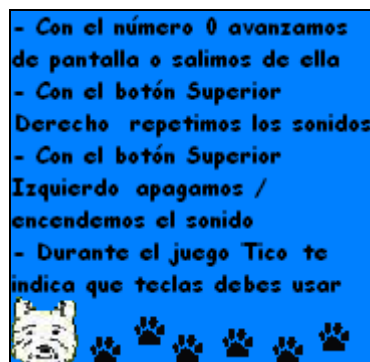


Imagen 22 .Ayuda

## 8. DESARROLLO PROYECTO JUEGO EDUCATIVO PARA APLICACIÓN MÓVIL

---

### 8.1 FASE DE ANÁLISIS

#### 8.1.1 Descripción general del Sistema

En este apartado se explica la descripción general del Sistema, es decir, la capacidad del sistema y sus limitaciones, así como una breve introducción a cerca de la arquitectura empleada.

Se expondrá el porqué de determinadas decisiones a la hora de desarrollar la aplicación y cómo es el transcurso del juego, ahora que ya hemos explicado las diferentes pantallas que aparecen en la aplicación.

##### 8.1.1.1 Capacidades generales

La aplicación intenta mostrar al usuario los diferentes peligros diarios a los que se puede enfrentar en el hogar.

Primeramente al usuario se le da la posibilidad a través de un menú de realizar diversas opciones.

Estas opciones son:

- \* *Activar / Desactivar Sonido:* El usuario tiene la opción de desarrollar el juego con una música ambiental pulsando esta opción en el menú o desarrollar el juego sin música.
- \* *Ayuda:* Opción del menú a través de la cual, se le explica de forma breve al usuario como manipular el teclado para poder desarrollar el juego.
- \* *Ranking:* En caso de que el usuario quiera acceder al histórico de puntos de sus partidas y de las de otros usuarios, a través de esta opción podrá ver a modo de clasificación las mejores puntuaciones existentes en la base de datos.
- \* *Jugar:* Con esta opción, el usuario iniciará una nueva partida, partiendo su puntuación con 0 puntos y pudiendo demostrar tantas veces quiera su destreza con la aplicación.
- \* *Salir:* Si el usuario no quiere ejercer ninguna opción de las anteriores descritas y desea abandonar la aplicación, a través de esta opción podremos cerrarla.

En caso de que el usuario elija la opción de jugar, se iniciará una nueva partida, mostrando inicialmente una pantalla con una breve introducción al juego, y posteriormente otra pantalla explicándole las acciones a las que se enfrenta el jugador cuando se muestre la pantalla en donde se desarrollan estas.

Entre pantalla y pantalla en las que el usuario desarrolla el juego, existe siempre una pantalla previa explicativa del suceso al que se enfrenta el jugador. Debido a que la velocidad de lectura de un niño de 8 años, no es la misma que la de un niño de 12, el usuario a través del intro (botón 0), será el que dé el paso de la pantalla que explica los sucesos a los que se va a enfrentar el usuario, a la pantalla propia en la que debe de realizar las acciones.

Por otro lado, en todas las pantallas donde el usuario elige las opciones que el cree que son correctas, (exceptuando las de Bonus) puede pasar que el usuario haya elegido las opciones correctas, en tal caso se mostrará una nueva pantalla aplaudiendo la correcta elección de las opciones (durante 2 segundos) y en caso de que haya tenido un error en la opción u opciones, se mostrará una pantalla que dará la oportunidad al usuario de repetir de nuevo la pantalla, pudiendo elegir en este caso las opciones correcta o de seguir a la siguiente pantalla de la aplicación.

Indicar que se realiza los aciertos, mostrando la pantalla de aplausos, pero por el contrario, en caso de fallar no se emite ningún sonido para no desilusionar al joven usuario.

Existen unas pantallas de Bonus, las cuales, se muestran durante un tiempo establecido de 3 segundos, en donde el usuario debe memorizar los objetos de la pantalla y contestar a la pregunta que se le hace después de haber pasado esos 3 segundos. En estas pantallas no se da la opción de poder volver a repetir el bonus por completo, pero al finalizar el bonus, sí que se muestra al usuario, las preguntas con las contestaciones propias de éste, así como las respuestas correctas en caso de que haya fallado previamente.

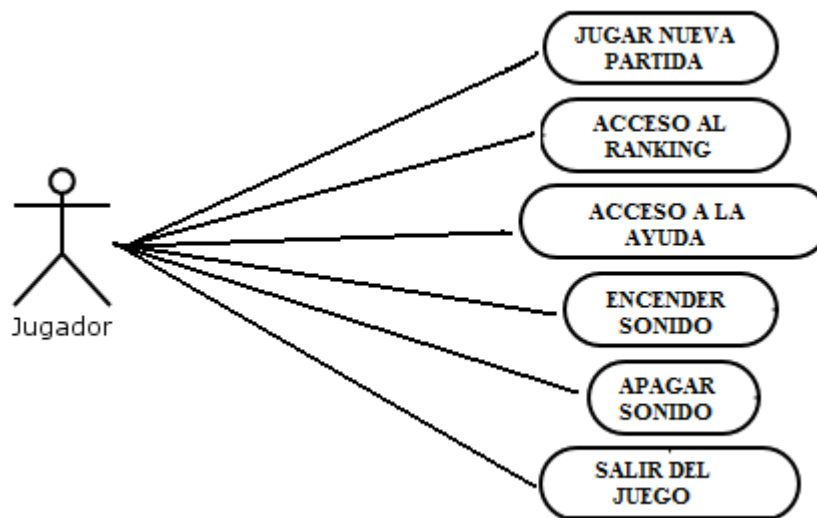
Cuando el usuario finaliza la partida, se visualiza la puntuación final y después de pulsar el intro, visualizamos de nuevo el menú.

#### **8.1.1.2 Diagramas de casos de uso**

A continuación y con el fin de aclarar el funcionamiento de la aplicación, se presenta los diagramas de casos de uso.

Los casos de uso representan la interacción entre el usuario y la aplicación educativa, mostrando la relación entre el videojuego y su entorno.

Se muestra un único escenario, en el cual, el usuario interactúa con el sistema.



*Imagen 23. Diagrama Casos de Uso*

#### **8.1.1.3 Restricciones generales**

En este apartado se incluyen las posibles limitaciones a las que puede verse ligada la aplicación, afectando al proceso de desarrollo y al resultado del propio videojuego.

Primeramente, me gustaría indicar que se trata de un proyecto llevado a cabo por una única persona, lo cual no quiere decir que se lleve a cabo con esfuerzo y constancia para que el resultado del videojuego sea el esperado por el usuario.

Por otro lado, hay que tener en cuenta, que la finalidad de dicho proyecto, no solo se trata de desarrollar un juego para móvil a través de la tecnología J2ME, si no que a su vez, y de forma paralela, intenta educar a los más pequeños de la casa, para prevenir riesgos que en su vida diaria les puede surgir dentro de su hogar. También, de forma paralela el juego intenta en el mayor grado posible dar y mostrar facilidades de interacción para niños con posibles deficiencias sensoriales. Por ello, el tamaño de las letras en las pantallas explicativas es de gran dimensión, los colores de los fondos de las pantallas intentan ser opuestos, a los colores de las letras, para que resalten más las letras, se da la opción de mostrar el sonido o por el contrario apagarlo. Los dibujos de las pantallas están formados por objetos sencillos para poder ser fácilmente asimilados.

Indicar, que la aplicación está hecha con la tecnología J2ME, debido a que un usuario de temprana edad, no suele tener un Iphone, o un móvil táctil, sino que suelen adquirir móviles baratos comprados por sus padres, o móviles que apartan sus padres o hermanos debido a que vayan a adquirir otros de mayor gama, es por eso, que la tecnología J2ME es la más indicada debido a que un juego de estas dimensiones, consumirá menos recursos en esta tecnología, que no desarrollándola con un sistema Androide para móviles táctiles, cuando lo normal es que en edad tempranas no se disponga de este tipo de móviles.

### **8.1.3 Características de los Usuarios**

Primeramente, destacar, que el proyecto consiste en una aplicación cuyo objetivo a parte de divertir a los más pequeños, tiene un carácter educativo, debido a que intenta que los más pequeños de la casa, asimilen como actuar ante determinados riesgos cotidianos.

Por tanto, los usuarios a los cuales va dirigida la aplicación se encuentran bien definido siendo éstos en su mayor parte niños con edades comprendidas entre 6 y 10 años que han comenzado a usar el teléfono móvil y poseen uno propio.

Debemos de tener en cuenta, que como media, a los 8 años es cuando muchos niños comienzan a usar los teléfonos móviles, y suele ser a estas edades cuando en el colegio se les comienza a dar una educación vial, por lo que, en este rango de edades, estaría comprendido el usuario ideal de la aplicación.

### **8.1.4 Requisitos de Usuario**

A continuación se presenta dos categorías de Requisitos, teniendo en cuenta las necesidades del joven usuario.

***Requisitos de Capacidad:*** Representan lo que necesitan los usuarios para poder desarrollar y divertirse con la aplicación.

***Requisitos de Restricción:*** Restricciones impuestas principalmente, por los posibles problemas de accesibilidad que nos podemos encontrar entre los más pequeños.

Cada requisito de usuario se presenta con una serie de atributos, de forma que añaden la información necesaria para su posterior seguimiento.

#### **Identificador:**

- **RU – C – xx** donde:

- **RU:** Indica que se trata de un requisito de usuario.
- **C:** Indica que se trata de un requisito de capacidad.
- **xx:** Número identificativo del requisito.
- **RU – R – xx** donde:
- **RU:** Indica que se trata de un requisito de usuario.
- **R:** Indica que se trata de un requisito de restricción.
- **xx:** Número identificativo del requisito.
- **Nombre:** Descripción general del requisito
- **Necesidad:** Los requisitos se clasificarán de acuerdo a su necesidad de la siguiente forma: Esencial, Deseable u Opcional.
- **Descripción:** Desarrollo textual del requisito.

#### 8.1.4.1 Requisitos de Capacidad

IDENTIFICADOR RU-C-01	
NOMBRE	Dispositivo Móvil
NECESIDAD	Esencial
DESCRIPCION	Para poder disfrutar de la aplicación será necesario disponer de un teléfono móvil

*Tabla 2. RU-C-01*

<b>IDENTIFICADOR RU-C-02</b>	
<b>NOMBRE</b>	Visualizar Ranking
<b>NECESIDAD</b>	Deseable
<b>DESCRIPCION</b>	Se mostrará los datos del fichero que contenga el Ranking de los jugadores

*Tabla 3. RU-C-02*

<b>IDENTIFICADOR RU-C-03</b>	
<b>NOMBRE</b>	Comenzar Nueva Partida
<b>NECESIDAD</b>	Esencial
<b>DESCRIPCION</b>	Se dará opción al usuario de poder iniciar una nueva partida para poder divertirse con la aplicación y asimilar los conceptos

*Tabla 4. RU-C-03*

<b>IDENTIFICADOR RU-C-04</b>	
<b>NOMBRE</b>	Ayuda
<b>NECESIDAD</b>	Deseable
<b>DESCRIPCION</b>	Se mostrará al usuario un sistema de ayuda de forma que pueda saber que realiza cada función proporcionada por la aplicación

*Tabla 5. RU-C-04*

<b>IDENTIFICADOR RU-C-05</b>	
<b>NOMBRE</b>	Salir
<b>NECESIDAD</b>	Esencial
<b>DESCRIPCION</b>	El usuario podrá abandonar la aplicación sin haber terminado la partida o bien una vez finalizada

*Tabla 6. RU-C-05*



#### 8.1.4.2 Requisitos de Restricción

IDENTIFICADOR RU-R-01	
NOMBRE	Conocimiento Lectura
NECESIDAD	Deseable
DESCRIPCIÓN	

*Tabla 7. RU-R-01*

IDENTIFICADOR RU-R-02	
NOMBRE	Accesibilidad Auditiva
NECESIDAD	Deseable
DESCRIPCION	

*Tabla 8. RU-R-02*

IDENTIFICADOR RU-R-03	
<b>NOMBRE</b>	Accesibilidad Cognitiva
<b>NECESIDAD</b>	Deseable
<b>DESCRIPCION</b>	

*Tabla 9. RU-R-03*

IDENTIFICADOR RU-R-04	
<b>NOMBRE</b>	Accesibilidad Visual
<b>NECESIDAD</b>	Deseable
<b>DESCRIPCION</b>	

*Tabla 10. RU-R-04*

<b>IDENTIFICADOR RU-R-05</b>	
<b>NOMBRE</b>	Accesibilidad en la Movilidad
<b>NECESIDAD</b>	Deseable y esencial un mínimo de movilidad que permita pulsar un botón
<b>DESCRIPCION</b>	

*Tabla 11. RU-R-05*

<b>IDENTIFICADOR RU-R-06</b>	
<b>NOMBRE</b>	Configuración Sonido
<b>NECESIDAD</b>	Esencial
<b>DESCRIPCION</b>	La aplicación permite configurar el volumen del sonido

*Tabla 12. RU-R-06*

IDENTIFICADOR RU-R-07	
NOMBRE	Idioma
NECESIDAD	Esencial
DESCRIPCION	El idioma de la aplicación será el castellano

*Tabla 13. RU-R-07*

## 8.2 FASE DE DISEÑO

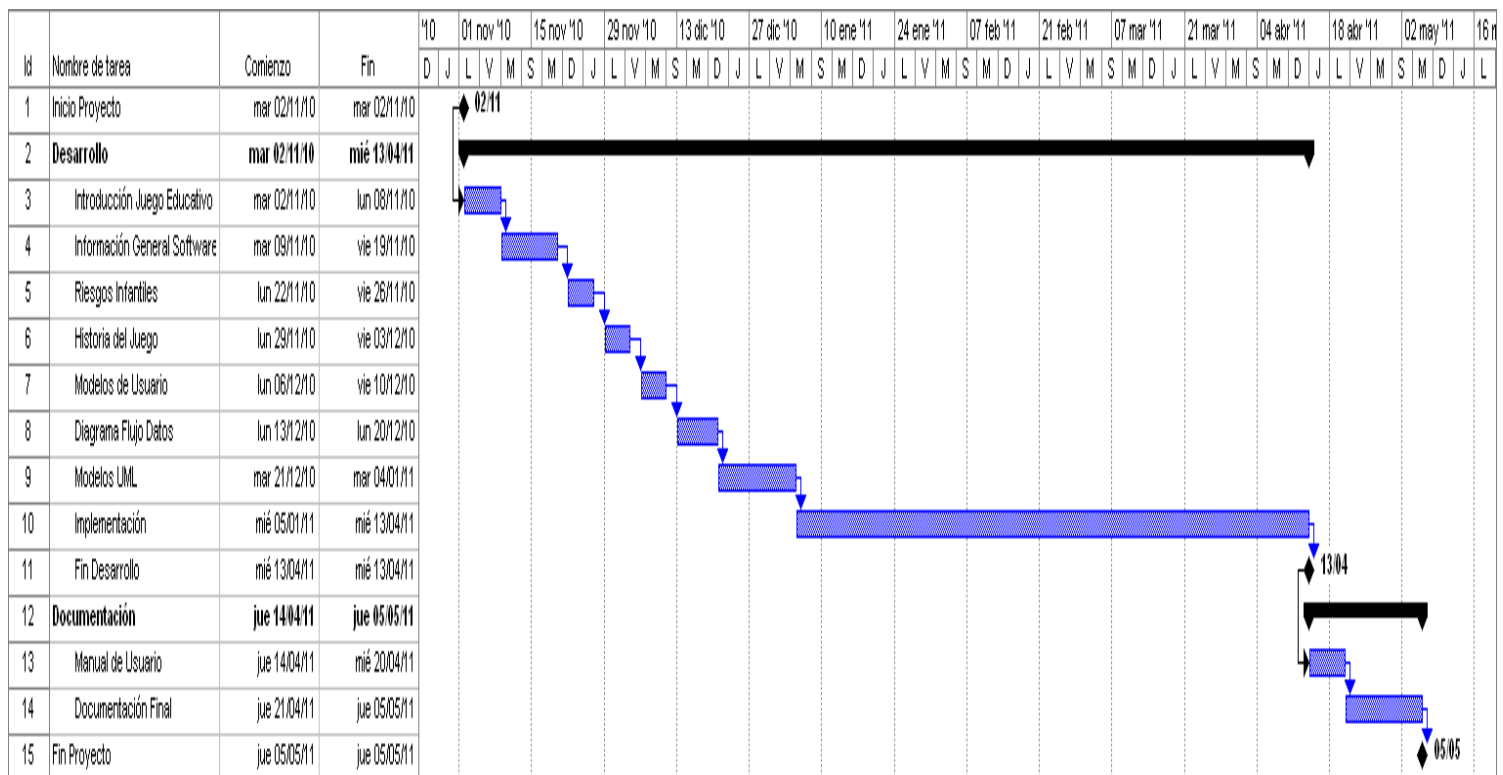
### 8.2.1 Fecha de inicio y fecha de fin

La fecha inicial para el comienzo del proyecto es el martes 02 de Noviembre de 2010. A partir de dicha fecha es cuando se comenzarán a imputar las horas de trabajo del personal asociado al proyecto.

La fecha de finalización del proyecto se ha fijado para el miércoles 11 de Mayo de 2011, día en el que quedará por cerrado el proyecto. Además el equipo de trabajo se compromete a entregar toda la documentación y software generado para la aplicación, de forma que el proyecto quede finalizado de forma satisfactoria.

A continuación se muestra el diagrama de Gantt, el cual, indica la planificación de cada una de las tareas que se van a llevar a cabo desde la fecha inicial indicada hasta el cierre del proyecto.

### 8.2.2.Diagrama de Gantt



*Imagen 24 Diagrama Gantt*

### 8.2.3 Estimación de Costes

Posteriormente al Diagrama de Gantt planificado para poco más de 6 meses de duración del proyecto, estableciendo el detalle de las tareas y su duración, se incluye a continuación un presupuesto detallado de los costes del proyecto.

Previamente, se especifica de forma detalla que tareas del proyecto serán designadas para cada recurso de los que se dispone.

Los recursos de los que va a disponer el proyecto serán:

- 1 Jefe de Proyecto
- 1 Analista
- 1 Analista Programador

PUESTO	TAREA	DIAS	HORAS DIA	TOTAL HORAS
<b>JEFE PROYECTO</b>	INTRODUCCIÓN AL PROYECTO	24	8	192
	ORGANIZACIÓN PROYECTO	4	8	32
	ANÁLISIS Y DISEÑO SISTEMA (50%)	24 / 2	8	96
	DOCUMENTACIÓN (50%)	20 / 2	8	80
<b>ANALISTA</b>	ANÁLISIS Y DISEÑO SISTEMA (50%)	24 / 2	8	96
	DOCUMENTACIÓN (50%)	20 / 2	8	80
<b>ANALISTA - PROGRAMADOR</b>	IMPLEMENTACIÓN	68	8	544

*Tabla 14. Recursos Proyecto*

La idea de utilizar un Analista Programador en vez de un programador, es debido, a que se prefiere contar con una persona con mayor experiencia debido a que será el único recurso que utilizemos para la implementación.

Después de hacer especificado en la anterior tabla los recursos y las actividades de cada recurso, a continuación mostramos la tarifa por horas de cada recurso y el importe total del coste del proyecto.

DESCRIPCIÓN	PRECIO UNITARIO HORA	TOTAL
<b>JEFE PROYECTO</b>	45	$400h * 45 = \mathbf{18.000}$ <b>euros</b>
<b>ANALISTA</b>	38	$176h * 38 = \mathbf{6.688}$ <b>euros</b>
<b>ANALISTA PROGRAMADOR</b>	30	$544h * 30 = \mathbf{16.320}$ <b>euros</b>
<b>APARATO MOVIL PRUEBAS</b>	-	<b>100 euros</b>

*Tabla 15. Costes Proyecto*

La cuantía de costes del proyecto asciende a 41.108 euros brutos, a los cuales debemos de añadirles el 18% I.V.A., por lo que el gasto total del proyecto asciende a 48.507,44 euros

#### 8.2.4 Diagrama de Clases

A continuación se muestra el diagrama de clases en alto nivel para dar una visión general de los conceptos que maneja la aplicación.

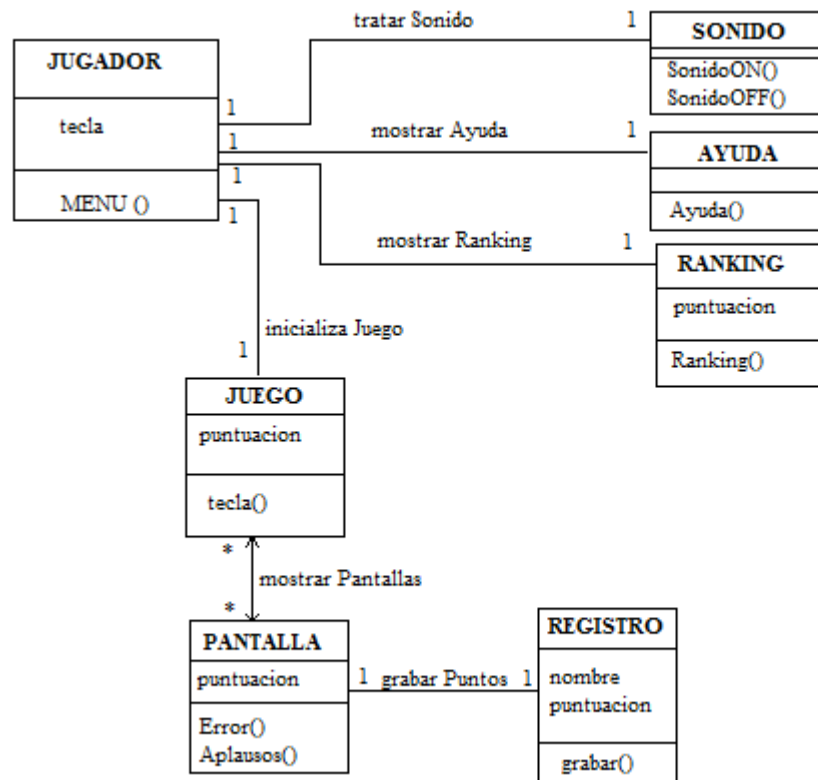


Imagen 25. Diagrama de clases

**Jugador:** Representa la interfaz de la herramienta.

**Sonido:** Clase a través de la cual se activa o desactiva el sonido

**Ayuda:** Mediante esta clase, se muestra la pantalla de ayuda para el desarrollo de la aplicación.

**Ranking:** Clase que visualiza el histórico de jugadores y puntuaciones obtenidas por ellos. Muestra las 5 mejores puntuaciones.

**Juego:** Es la clase que funciona como controladora de la aplicación, recibiendo la tecla pulsada y enviando a la pantalla correspondiente a la aplicación.



**Registro:** Clase a través de la cual registramos el nombre del jugador y su puntuación correspondiente si es mejor que las 5 puntuaciones que constan en la base de datos.

### 8.2.5 Diagrama de Secuencias

En el diagrama de secuencias representamos las diferentes formas de interactuar del usuario con la aplicación.

El diagrama nos muestra que desde que el usuario inicia la aplicación a través del menú puede acceder a tres acciones diferentes como son el acceso al ranking, el acceso a la ayuda o el acceso al juego.

En caso de acceder al ranking o a la ayuda, vemos que la única actuación posible del usuario es volver al menú, así como encender o apagar el sonido de la aplicación.

Si el usuario decide acceder a jugar, a lo largo de las pantallas con las que se encuentra en el juego podemos ver a través del diagrama que podrá interactuar apagando o encendiendo el sonido, volviendo a repetir la pantalla en caso de que no haya respondido correctamente o en caso de que todo haya ido bien, acceder a la posterior pantalla.

En última instancia, el diagrama nos muestra la secuencia de juego en donde el usuario graba sus datos en la base de datos pudiendo encender o apagar el sonido como única forma de interactuar con la aplicación.

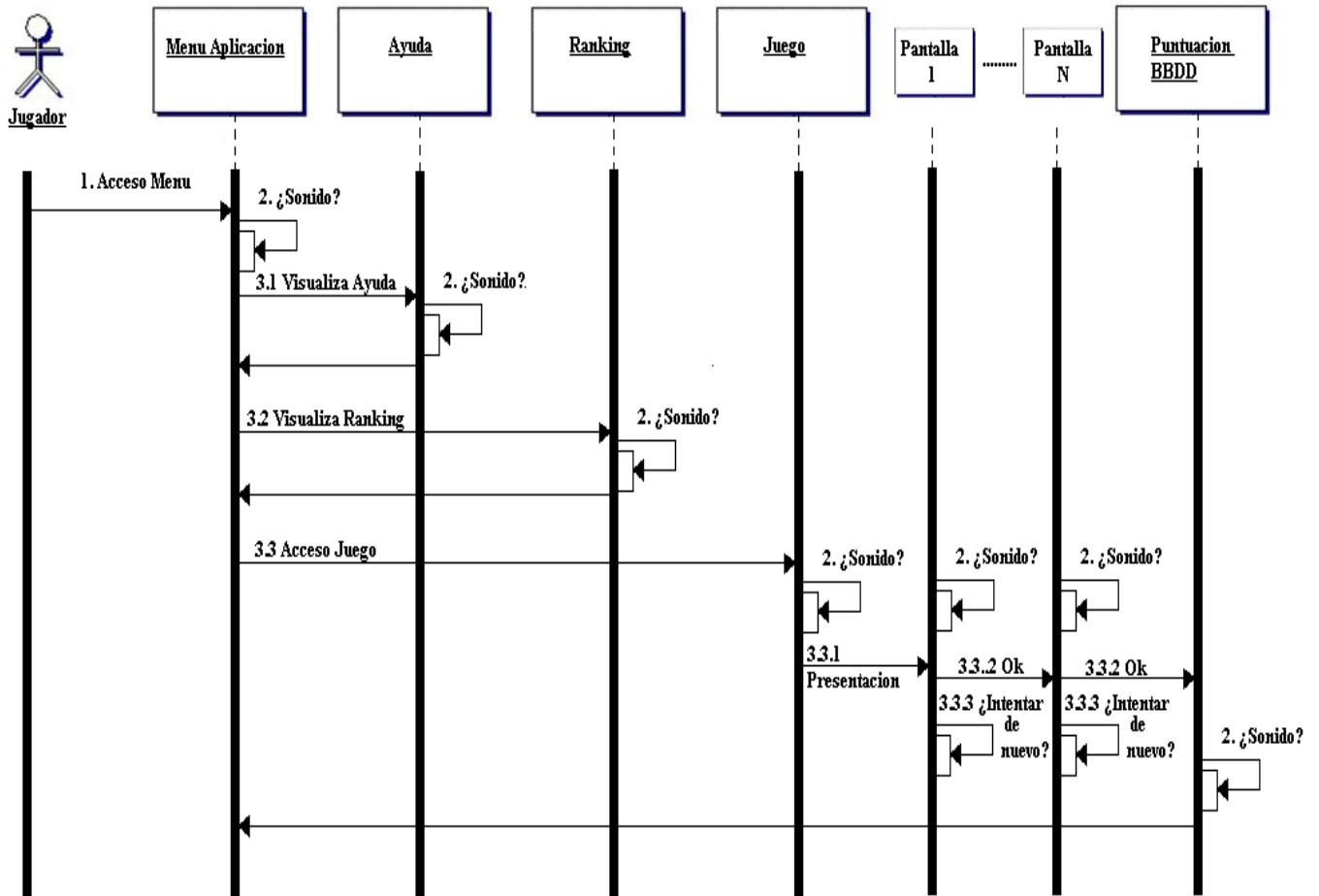


Imagen 26. Diagrama de Secuencias

## 9. VALIDACIÓN

---

Es importante realizar una fase de pruebas una vez finalizado el desarrollo e implementación del sistema. Con ellas comprobamos las prestaciones del proyecto y que el funcionamiento es el deseado. Nos ayuda a garantizar el cumplimiento de los requisitos impuestos y encontrar posibles errores en la aplicación.

Estas pruebas las basamos en:

**Acceso a la aplicación:** Accedemos a la aplicación y nos cercioramos de que no surge ningún error o warning que dificulte la funcionalidad del transcurso de la aplicación

**Acceso a “Menu”:** Comprobamos que tanto al inicio de la aplicación como al finalizar una partida, el juego parta o reanude la aplicación desde el menú de inicio, comprobando que solo interactúan las teclas que presentan un acceso a alguna funcionalidad del juego.

**Acceso a “Salir”:** Accedemos a la opción del menú de Salir, verificando que con ambas opciones , la de salir o no salir, el juego responde como esperamos y que ante la pulsación de cualquier otro botón que no tenga que ver con la funcionalidad de esta pantalla, la aplicación no ejecuta ninguna acción

**Acceso a “Ayuda”:** Accedemos a la opción Ayuda del menú de la aplicación, verificando que la aplicación muestra la pantalla correspondiente y que tan solo el botón de volver es el que interactúa en esta pantalla.

**Acceso a “Ranking”:** Accedemos a la pantalla de Ranking para ver que en caso de no existir puntuaciones no visualiza ninguna, en caso de que existan varias puntuaciones nos las visualiza de mayor a menor puntuación y también comprobamos que en caso de que existan en la Base de Datos más de 5 puntuaciones guardadas nos visualice las cinco puntuaciones con los resultados más altos.

**Acceso a “Jugar”:** El acceso a esta parte de la aplicación nos lleva el mayor tiempo y número de pruebas puesto que la acción “jugar” conlleva muchas pantallas y en cada una de ellas hemos tenido que comprobar que en caso de no contestar lo correcto se nos visualiza la pantalla que nos da la opción de volver a intentarlo. Por otra parte, en caso de contestar lo correcto hemos verificado que se nos visualiza la pantalla de felicitación y que posteriormente se visualiza la pantalla de la prueba posterior correspondiente.

En cada pantalla se ha comprobado que se puede volver a intentar un número indefinido de veces la resolución de la prueba, así como, que solamente interactúan los botones que en cada pantalla se muestra como posibles respuestas a las preguntas que se nos plantean.

**Acceso a “Grabar Datos”:** En esta pantalla, que pone fin al juego, hemos verificado que con todas las puntuaciones posibles y animales posibles para ser elegidos y grabar nuestros datos, estos, se graban correctamente en la Base de Datos, así como su audición y visualización de datos es correcta tras haber sido grabados por el usuario.

**Acceso a “Sonido”:** El acceso al sonido, depende de si lo queremos encendido o apagado. La validación correcta del funcionamiento de esta acción se ha elaborado llevando a cabo una exhaustiva corroboración de que pantalla a pantalla independientemente de si es la pantalla de *Salir*, o de si hemos accedido a la *Ayuda*, o hemos querido visualizar el *Ranking*, o acabamos de iniciar el juego o finalizar una partida y nos encontramos en el *Menú*, por supuesto en todas y cada unas de las pantallas del *Juego*, o finalmente si hemos finalizado la partida y hemos accedido a *Grabar dato*, en todas y cada una de ellas se han llevado a cabo pruebas de visualizar la pantalla sin sonido, o visualizarla con sonido, iniciar la visualización de la pantalla sin sonido y luego introducirlo, o viceversa, visualizando la pantalla con sonido y posteriormente quitándolo, y de igual modo, hemos comprobado en cada pantalla que funcionaba correctamente el volver a escuchar un sonido de cualquier pantalla que hubiera finalizado.

## 10. CONCLUSIONES Y LÍNEAS FUTURAS

---

Hoy en día existen multitud de juegos educativos que ayudan tanto en las escuelas infantiles como en los colegios a que los más pequeños aprendan de una forma más amena y divertida e intenten asimilar de esta forma nuevos conocimientos que puedan poner en práctica posteriormente en su día a día.

En Internet también encontramos diversos accesos a aplicaciones educativas para los más pequeños puesto que cada día con menor edad aprenden a utilizar los ordenadores y saben manejar Internet y dichos ordenadores con mucha destreza.

En móviles también existen juegos que ayudan a que los niños asimilen nuevos conocimientos, pero el número de estos juegos es menor que en los dos casos anteriormente descritos.

Por otra parte, aparte de que un juego eduque como hemos nombrado anteriormente, en este proyecto se intenta integrar y que se integren aquellos más desfavorecidos y que presentan una disminución sensorial y que no siempre pueden acceder a todos los servicios, y a toda la oferta que nos propone la industria del móvil o de Internet.

A través de este juego educativo, aparte de intentar enseñar a los más pequeños los peligros con los que se puede encontrar en su rutina diaria en su hogar, se intenta también que aquellos que padezcan una disfunción sensorial incluso auditiva o visual puedan de igual forma establecer una interacción con el móvil y puedan adquirir de igual forma esos conocimientos que el juego intenta transmitir.

Para desarrollar esta aplicación, hemos utilizado como lenguaje de programación J2ME de Java y como arquitectura hemos usado Eclipse. Para poder realizar la interacción con el móvil se ha usado el simulador Wireless Toolkit, el cual nos ha permitido ir viendo poco a poco como se iba consagrando este juego didáctico y educativo que a través de RMS nos ha permitido de igual forma dejar grabado los resultados de las puntuaciones que obtienen los más pequeños al desarrollar una partida.

En cuanto a los estudios de accesibilidad y usabilidad, he llevado a cabo diferentes medidas a la hora de diseñar el juego para que pueda ser lo mayormente accesible a niños con deficiencias sensoriales y pueda mostrar el juego la mayor usabilidad posible dentro de las limitaciones que puedan tener aquellos niños con deficiencias.

Para ello, he realizado diversas acciones como las que se describen a continuación para en la mayor medida posible hacer al juego lo mayormente accesible posible y con mayor usabilidad.

\* Cualquier parte del juego puede ser leída o escuchada, lo cual me parece fundamental para que una persona con deficiencias auditivas se le pueda asegurar que puede jugar con total seguridad al juego ya que podrá leer toda la información y las acciones correspondientes al juego, así como, una persona con problemas de visión, a través de un teclado de móvil que contenga la traducción de sus números al sistema braille, se asegura a través del sonido, el poder llevar a cabo una partida con el juego e incluso guardar su puntuación y observar el ranking de sus puntos.

\* La forma en que el jugador guarda sus puntos de la partida es indispensable que sea a través de, en este caso, unos animales, para así poder guardar esos sonidos de animales en el desarrollo del juego y poder utilizarlos para que aquellas personas con problemas visuales puedan guardar sus datos y puedan igualmente escuchar sus puntuaciones.

\* Tras haber mostrado el juego a varios niños y darme cuenta de la diferencia de velocidad entre un niño de 8 años y uno de 12 años, se aplica la obligatoriedad de tener que pulsar el botón 0 del móvil para poder pasar a la siguiente pantalla, debido a que si pudiéramos las pantallas en las que explican el juego con un temporizador, para los niños pequeños podría ser poco tiempo para leerlo y para los más mayores podría resultar pesado tener que esperar ciertos segundos tras leer la pantalla para pasar a la posterior pantalla.

\* Otra de las premisas que se ha intentado llevar a cabo en el juego ha sido el contraste de colores para un mejor entendimiento y mayor accesibilidad para personas con pequeñas deficiencias, así como un trazo de los dibujos claro y simple para un fácil entendimiento de lo que se quiere expresar en cada pantalla.

\* Dar la posibilidad de repetir la pantalla o el hecho de felicitar al niño cuando ha acertado plenamente en los resultados de la pantalla, son dos tipos de acciones que por un lado, en caso de haber acertado, se refuerza su autoestima felicitándolo, y en caso de haber fallado se le premia de igual forma con una segunda e infinitas oportunidades para que pueda conseguir finalmente hacerlo bien.

\* El evitar el scroll en las pantallas, premia una usabilidad más sencilla para aquellos pequeños con problemas sensoriales, puesto que realmente estamos evitando que tengan que realizar una acción más para desarrollar el juego y que el hecho de usar el scroll no les otorga ningún aprendizaje sobre lo que les estamos intentando inculcar.

A título personal, he aprendido una nueva forma de trabajar y de programar que es con el lenguaje J2ME y el uso del simulador Wirless Toolkit, que anteriormente no conocía, y he comprobado que el lenguaje no presenta grandes dificultades, pero el emulador si presenta inconvenientes como solo poder aplicar sonidos WAV a través de J2ME y de eclipse, lo cual ha hecho que el proyecto haya cogido bastante peso en cuanto a tamaño y numero de archivos de sonido, debido a que no permitía sonidos MP3 que ocupan un tamaño bastante menor.

Las líneas futuras que se podrían emplear para mejorar este juego educativo podrían ser diversas y variadas. A continuación detallamos como podríamos mejorarlo y que podríamos añadir.

- \* Lo más importante sería poder desarrollarlo con una arquitectura o software que soportara sonidos MP3, puesto que el mayor problema que me he encontrado es, que Eclipse solo soporta formato WAV de sonido, el cual es hasta 10 veces superior en peso de Kb a un archivo MP3. Con esto conseguiríamos evitar la ralentización que sufre hoy en día el juego cuando está activado el sonido.

- \* Para mejorarlo y hacerlo más educativo se podría aumentar la historia del juego estableciendo nuevas pantallas que educaran a los niños en otros aspectos como podría ser la prevención en los riesgos naturales que pueden surgir en la vida.

- \* Se podría establecer una conexión wifi entre varios teléfonos y dotar al juego de la posibilidad de jugar a varios jugadores a la vez.

- \* Podríamos incluir la posibilidad de jugar en diferentes idiomas, ya que hoy en día se intenta integrar el bilingüismo en las escuelas.

- \* Otra opción para mejorar el juego en el futuro sería realizarlo para móviles con pantallas táctiles, aunque son muy pocos los niños que hoy en día usan móviles con pantalla táctil y por otro lado este tipo de pantalla dificulta la accesibilidad para niños con deficiencias visuales.

## GLOSARIO

---

<b>J2ME</b>	Java 2 Micro Edition
<b>GNU GPL</b>	General Public License
<b>JVM</b>	Java Virtual Machine
<b>JIT</b>	Just In Time
<b>Gcj</b>	GNU Compiler for Java
<b>VM</b>	Virtual Machine
<b>OSGi</b>	Open Services Gateway Initiative
<b>OAK</b>	Roble (Programacion Roble)
<b>VCR</b>	Videocasete Recorder
<b>PDA</b>	Personal Digital Assistant
<b>LCD</b>	Liquid Cristal Display
<b>J2SE</b>	Java 2 Standard Edition
<b>JDK</b>	Java Development Kit
<b>MIDP</b>	Mobile Information Device Profile
<b>AMS</b>	Application Management Software
<b>JAD</b>	Java Decompiles
<b>RMS</b>	Record Management System
<b>JDBC</b>	Java Database Connectivity
<b>SQL</b>	Structure Query Language
<b>CLDC</b>	Connected Limited Device Configuration
<b>API</b>	Application Programming Interface



<b>JTWI</b>	Java Technology Wireless Industry
<b>MSA</b>	Mobile Service Architecture
<b>WMA</b>	Windows Media Audio
<b>MMAPI</b>	Mobile Media API
<b>SATSA</b>	Security and Trust Services API
<b>LAPI</b>	Livelihoods Application Programmer Interface
<b>CHAPI</b>	Content Handler Application Programmer Interface
<b>SIP</b>	Session Initiation Protocol
<b>SVG</b>	Scalable Vector Graphics
<b>PAPI</b>	Process Application Programmer Interface
<b>AMMS</b>	Advanced Multimedia Supplements
<b>JAR</b>	Java Archive
<b>JAD</b>	Java Decompiler
<b>IDE</b>	Integrated Development Environment
<b>WAI</b>	Web Accessibility Initiative
<b>WCAG</b>	Web Content Accessibility Guidelines
<b>ATAG</b>	Authoring Tool Accessibility Guidelines
<b>CC</b>	Closed Caption
<b>MP3</b>	MPEG(Moving Picture Experts Group) Audio Layer III

## BIBLIOGRAFÍA

---

\*Brand, R. (1995). Punished by Rewards? Septiembre 1995

\*Cabrera, P., (Septiembre, 2010) . Con los juegos también se aprende. Accesible en: <http://grou.ps/aprendiendojuntos/blogs/item/con-los-videojuegos-tambien-se-aprende>

[Consultado, Septiembre 2011]

\*Carmona,C., Millán,E. (2006). MITO. Diseño y Evaluación de un Juego Educativo para Ortografía. Departamento de Lenguajes y Ciencias de la Computación Universidad de Málaga.

\*CiberAula. (2010). Accesible en: [http://java.ciberaula.com/articulo/introduccion\\_j2me/](http://java.ciberaula.com/articulo/introduccion_j2me/)

[Consultado, Marzo 2012].

\*Compute-rs. (2011). Accesible en: <http://www.compute-rs.com/es/consejos-2890110.htm> [Consultado, Octubre 2011]

\*Enlaces, Centro de Educación y Tecnología Chile, Plan Maestro Capacitación Año 2, Accesible en: [http://www.c5.cl/redenlaces/Recursos/Manuales/Evaluacion\\_de\\_Recursos\\_Educativos\\_Digitales.pdf](http://www.c5.cl/redenlaces/Recursos/Manuales/Evaluacion_de_Recursos_Educativos_Digitales.pdf) [Consultado, Septiembre 2011]

\*García Galán, S. Universidad de Jaén, Dpto Telecomunicaciones. Área de Ingeniería Telemática.

\*García Serrano, A. Accesible en: [www.agserrano.com/publi.html](http://www.agserrano.com/publi.html) [Consultado, Septiembre 2011]

\*Gosling, J. (2012)

Accesible en: [http://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

[Consultado, Marzo 2012].

\*Jiménez Rodríguez, E.(2006). I+E Revista Digital “Investigación y Educación” Vol III, Sevilla.

\*Mairena, J., (2009) Videojuegos Accesibles por qué y cómo hacerlos. 1ªEdición. Mayo 2009

\*Oracle (2012). Accesible en: <http://www.oracle.com/technetwork/java/overview-138293.html> [Consultado, Marzo 2012].

\*Orta, E., Cerezo, L., Hilardo, J., Figuera, L., Hernández, J.C., Briceño, R. Blog Software Educativo. Materia: Informática educativa (2008)

\*Pontificia Universidad Católica de Chile. Centro de Informática Educativa. Facultad de Educación. Facultad de Ingeniería. Software Educativo (2000)

Accesible en: <http://www.docstoc.com/docs/19502615/Clasificación-de-Software-Educativo> [Consultado, Septiembre 2011]

\*Sanz-Troyano, E., Torrente, J., Moreno-Ger, P., Fernández-Manjón, P., Dpto de Ingeniería del Software de Inteligencia Artificial. Facultad de Informática. Universidad Complutense Madrid

\*Zurita, G., Sánchez, J., Nussbaum, M.,(1999). Memorias Tise 99, Departamento de Ciencia de la Computación, Universidad de Chile, Pontificia Universidad Católica de Chile.

## **1. MANUAL DE USUARIO**

### **1.1. Introducción**

En este apartado se incluye un manual de usuario de la aplicación, desarrollado de forma que se ofrezca una guía para poder jugar aprovechando su jugabilidad.

Tiene como principal objetivo prestar ayuda al usuario final del juego, de forma que éste pueda utilizar la aplicación con éxito. En los puntos que siguen se explicará con detalle la instalación del juego en un dispositivo móvil, así como la creación del paquete Java a través del eclipse que nos permite instalar el juego en el móvil.

Primeramente en este manual, indicamos los lectores a los que va dirigida el juego educativo, especificando las características de estos. Posteriormente indicaremos el propósito u objetivo del manual y a continuación se explicará cómo manejar el manual de forma que se pueda obtener los mejores resultados.

### **1.2. Lectores a los que va dirigido**

Este manual está pensado para los posibles usuarios del juego educativo desarrollado. El usuario base, consiste en un niño de edad comprendida entre 8 y 12 años, que puede presentar alguna dificultad auditiva o visual y que tenga curiosidad por descubrir aquellos peligros a los cuales puede hacer frente en su vida cotidiana en el hogar.

### **1.3. Propósito**

El propósito de este documento es presentar la ayuda necesaria al usuario, de forma que éste pueda desempeñar una o las partidas que desee con el juego. Se explicará las posibles acciones a tomar en cada pantalla.

### **1.4. Como utilizar este manual**

El presente manual se corresponde con una ayuda al usuario para su consulta. La forma correcta de leerlo sería de principio a fin, de manera secuencial y prestando atención a cada uno de los apartados que forman parte del documento, ya que todos ellos aportan en mayor o menor medida información que debe ser tomada en cuenta por el usuario.

Para el correcto funcionamiento de la aplicación, es muy conveniente prestar atención a la forma de instalación.

### 1.5. Creación paquete

La creación del paquete Jar del proyecto, es intangible para el usuario, pero mostramos a continuación, la forma de empaquetar el proyecto para poder ser instalado posteriormente en un dispositivo móvil.

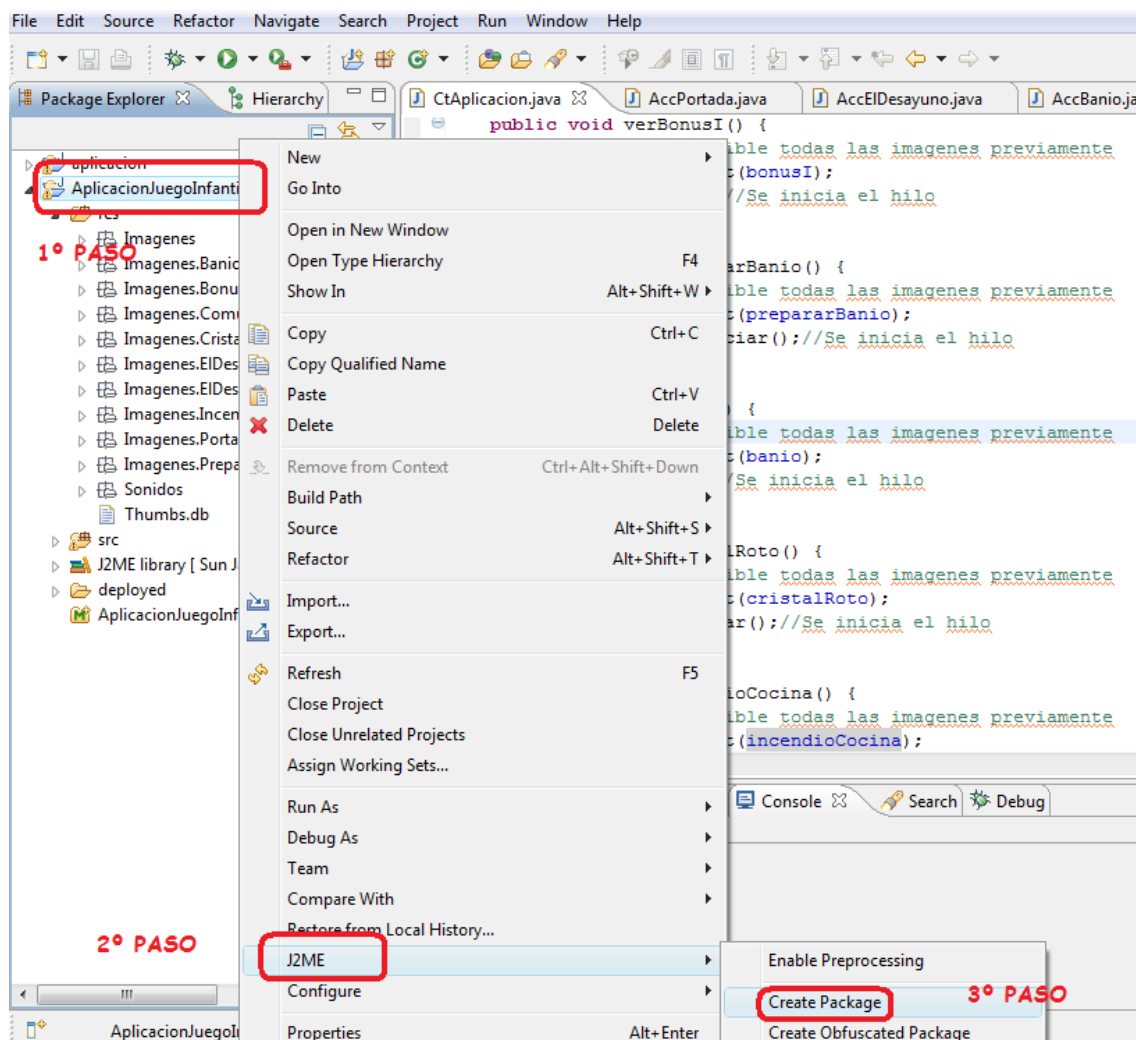
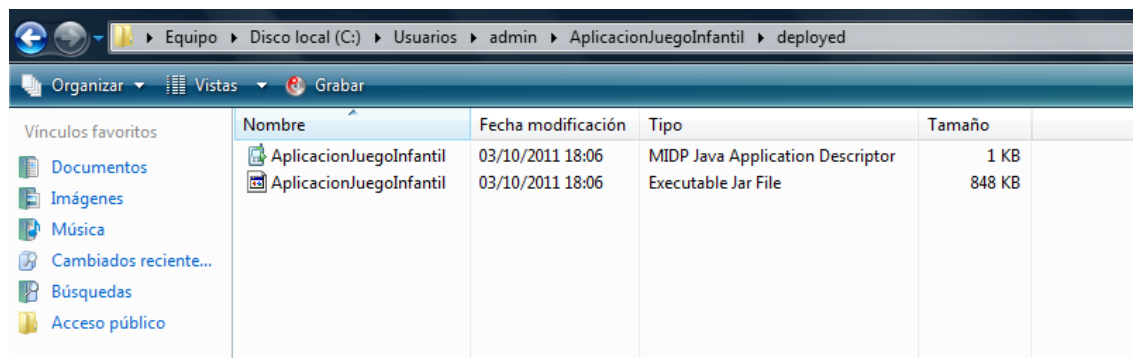


Imagen 27. Crear paquete Jar

Para la creación del paquete Jar seguimos tres pasos inicialmente.

- PASO 1: Sobre el paquete que engloba el proyecto, pulsamos el botón derecho del ratón para desplegar el menú de opciones.
- PASO 2: Elegimos la opción de J2ME para visualizar las opciones propias de su submenú.
- PASO 3: Dentro del submenú elegimos la opción “Create Package”.



*Imagen 28. Archivos Jar*

Finalmente, tras la generación del paquete, en la carpeta “Deployed” del proyecto se generarán dos archivos. Uno de los 2 archivos, tendrá extensión Jar, es decir, como podemos ver en la anterior imagen, su Tipo sera “Executable Jar File”, que será el fichero que copiaremos y pegaremos posteriormente en la carpeta de nuestro dispositivo móvil para poder instalar el juego educativo.

### **1.6. Instalación archivo Jar en dispositivo móvil**

Debido a la gran variedad de marcas de móviles existentes, en el manual indicaremos a través de notaciones generales como instalar de forma genérica un archivo Jar en un móvil.

- PASO 1: Para transferir el archivo jar al teléfono móvil es necesario en primer lugar, configurar e instalar un software en nuestro ordenador. Este software normalmente viene incluido en el mismo kit de compra del teléfono móvil a través de un CD. Si el mismo no viene incluido deberemos descargarlo desde la página oficial de la marca del dispositivo móvil indicando su modelo

- PASO 2: En segundo lugar, instalaremos en nuestro ordenador el software descargado de nuestro móvil. Esta instalación será muy breve y sencilla debido a que deberemos de pulsar por lo general el botón siguiente del paquete de instalación durante todo el proceso y a lo sumo, deberemos de indicar el modelo de nuestro dispositivo móvil.

- PASO 3: Para transferir el archivo jar (juego java) al teléfono móvil necesitaremos un cable de conexión entre el PC y el móvil (normalmente viene con el kit de compra), si no disponemos del cable podremos comprarlo en cualquier tienda de teléfonos móviles verificando siempre que sea compatible para nuestro modelo de móvil. Una vez conectados ambos equipos (móvil y ordenador) por medio del cable usb, el software instalado deberá detectar automáticamente que el móvil ha sido conectado al ordenador, si esto

no fuera así ejecutaremos directamente el software instalado desde el icono de acceso o desde el menú inicio buscando una opción en la aplicación que nos permitiera instalar aplicaciones en el móvil. Para comenzar a jugar solo tendremos que ir a aplicaciones/juegos (lee el manual de instrucciones de tu móvil ya que para cada modelo el lugar donde se aloja el juego es relativamente distinto), busca el archivo instalado y aceptar.

### **1.7. Requisitos técnicos dispositivo móvil**

Para poder hacer uso de la aplicación, necesitaremos de un dispositivo móvil que permita instalar aplicaciones java y que disponga de sonido. Es indiferente si contiene tarjeta de memoria externa, pues la aplicación puede instalarse en la memoria interna del teléfono, siempre que se disponga de capacidad para ello.

Debido al tema de accesibilidad, es más conveniente que el teclado del que disponga el teléfono no sea táctil, para facilitar a las personas con visibilidad reducida su manejo y puede ser no óptimo su uso en móviles táctiles, sí no se visualiza el teclado a la hora de estar desarrollando el juego.

### **1.8. Descripción funcional**

A continuación se describe las diversas opciones que muestra el juego y con las que el usuario puede interactuar.

La función principal de la aplicación es poder jugar una partida al juego, aunque el usuario puede acceder a la parte de menú que describe una pequeña ayuda sobre cómo jugar al juego o al ranking de puntuación del juego. También puede activar o desactivar el sonido del juego.

#### **1.8.1 Menú Juego**

La primera interacción con el juego una vez arrancado es la visualización del menú.

A través de este menú podemos acceder a las diferentes acciones que nos muestra el juego.

Cada acción nos llevará a otra pantalla que nos pida la verificación de la acción, como es el caso de querer salir del juego, o bien nos mostrará la pantalla propia de la acción elegida, como sucederá en caso de que elijamos “Ayuda”, “Ranking” o “Jugar”.

En caso de elegir “Sonido On” ,”Repetir” o “Sonido Off” las acciones se ejecutarán directamente sin mostrarse ninguna pantalla de verificación de la acción.



*Imagen 29. Menú*



### **1.8.2. Jugar**

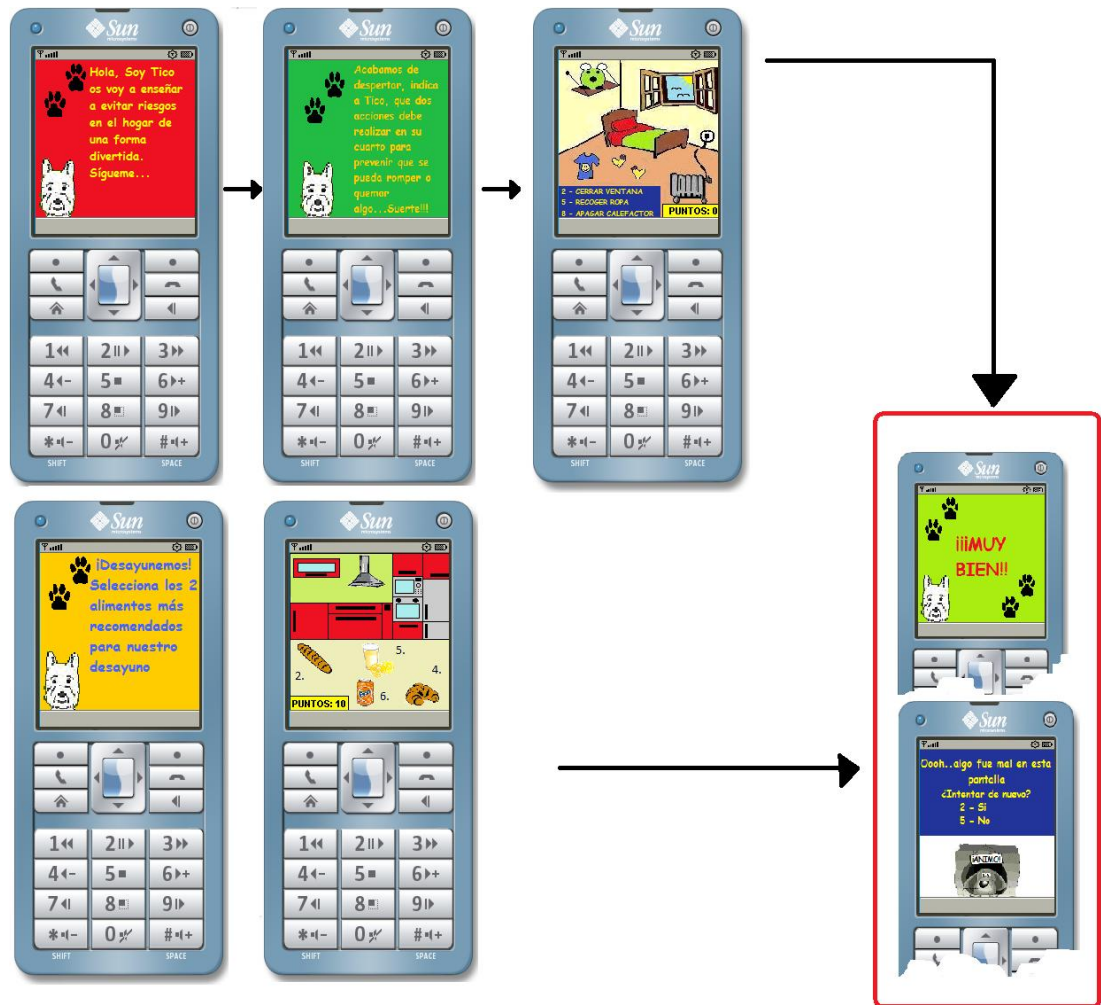
En caso de elegir la opción “Jugar” empezaremos a jugar al juego educativo desarrollado.

Las dos primeras pantallas del juego, que se muestran a continuación, muestran en primer lugar una descripción de la acción a desarrollar en la pantalla del juego, y posteriormente la pantalla de acción del juego.

Una vez el usuario haya interactuado en el juego, puede suceder que el usuario haya acertado las opciones correctas, con lo cual, se mostrará la pantalla “Muy Bien” acompañada de un sonido que transmite unos aplausos, o puede que el usuario no haya acertado las opciones, por lo cual, se mostrará la pantalla que le dará la opción de pasar a la siguiente pantalla o de volver a intentar de nuevo la pantalla.

Debe saberse que de cada pantalla explicativa, podremos dar paso a la siguiente pantalla pulsando el botón “0”.

Por otra parte, después de elegir las opciones que el usuario cree las acertadas, y transcurridos 3 segundos, se mostrará la pantalla de “Volver a intentar” o de “Muy Bien”, dependiendo de si se ha acertado o no.



*Imagen 30. Pantallas I*

De la pantalla “Volver a intentar” el usuario interactuará a través del botón “2” si quiere volver a intentarlo o con el botón “5” si quiere pasar a la siguiente pantalla.

De la pantalla “Muy Bien”, el juego tras 3 segundos, mostrará la siguiente pantalla.

Tras estas pantallas, el juego llega a un Bonus.

En este bonus, el juego visualizará previamente una pantalla de presentación, posteriormente aparecerá una pantalla de la cual el usuario deberá memorizar cada dibujo con el número que lo identifica y tras esa pantalla se visualizará una pregunta cuyo jugador deberá responder en base a los dibujos memorizados previamente.

Así, hasta en un total de 4 situaciones diferentes, que se le plantean al usuario.

Posteriormente, una vez contestadas a todas las preguntas del bonus, se visualizará correlativamente una pantalla con la solución y resultado correcto o incorrecto de cada una de las 4 preguntas que se formulan en el bonus.

Estas pantallas que muestran el resultado, se van visualizando una a una pudiendo pasar de pantalla a pantalla pulsando el botón “0”.

Come vemos en la imagen de la siguiente pantalla, el bonus sigue esta secuencia de imágenes, empezando por la pantalla de presentación y posteriormente mostrando las pantallas que nos darán la puntuación en el bonus. Finalmente, como muestra la imagen, aparecerán otras 4 pantallas correlativas que nos muestren el acierto o fallo a cada una de las preguntas del bonus.

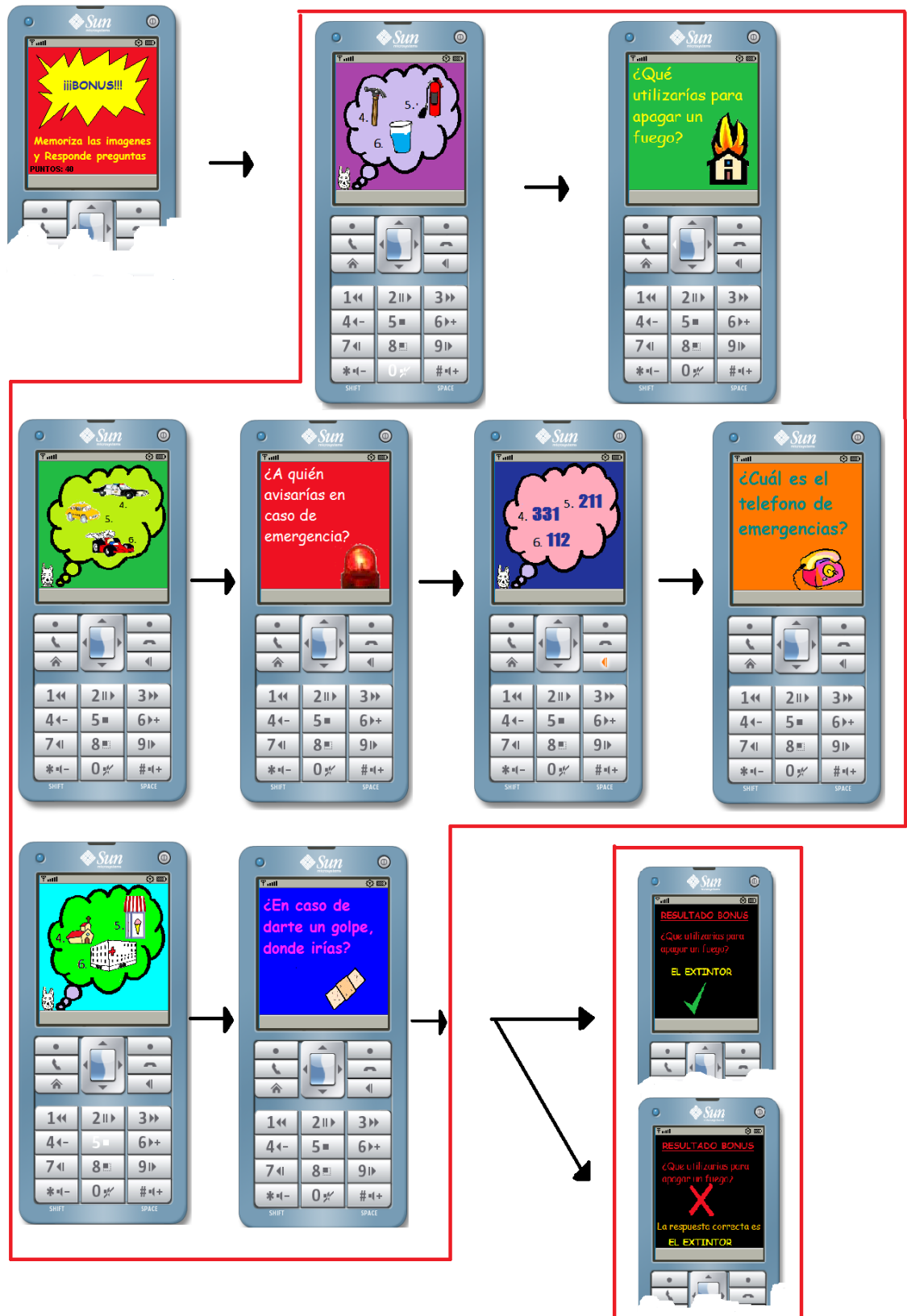


Imagen 31. Pantallas II

## Aplicación Educativa para aparatos móviles sobre los riesgos infantiles

Víctor Alonso Miranda

Posteriormente a estas pantallas de Bonus, el juego continua con otra serie de pantallas similares a las previas al Bonus, pero estableciendo nuevos retos al jugador.

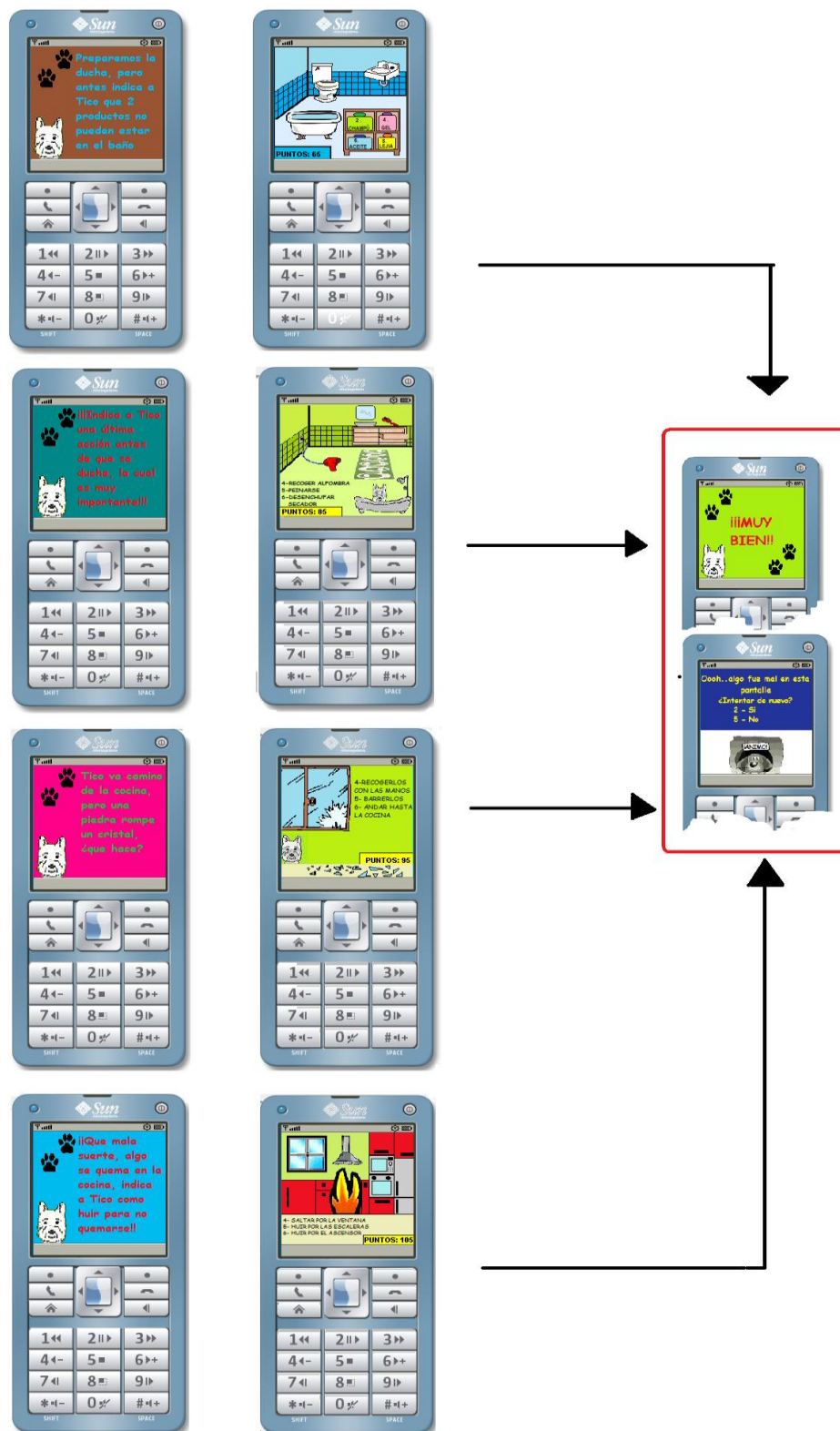


Imagen 32. Pantallas III

En esta sucesión de pantallas anteriormente expuestas, la jugabilidad es exactamente igual que a la del primer grupo de pantallas del juego expuestas (Imagen 4.Pantalla I).

Una vez el usuario ha finalizado el juego, debido a que ha desarrollado todas las pruebas que en él se exponen, podrá guardar su puntuación.

Debido a que la aplicación intenta ser un instrumento equitativo para los niños con o sin deficiencias sensoriales, se establece un sistema de grabación de puntuación auditivo estableciendo como jugadores unos determinados animales, de tal forma que un niño con deficiencias auditivas pueda escuchar con que animales puede guardar su puntuación.

En esta primera pantalla, el usuario elige el animal con el que desea guardar su puntuación.



*Imagen 33. Elección animal Puntuación*

En esta segunda pantalla, el usuario es informado de los puntos que ha obtenido y del animal con el que ha guardado su puntuación.



*Imagen 34. Datos Puntuación*



### 1.8.3 Ranking

En la opción del menú de Ranking, el usuario accede al Ranking del juego pudiendo visualizar y escuchar las 5 máximas puntuaciones obtenidas por los diferentes usuarios que han jugado una partida con el juego.



*Imagen 35. Ranking*

#### 1.8.4 Ayuda

En la opción del menú de Ayuda, el usuario accede a una pantalla donde se le muestra una pequeña ayuda del manejo de los botones que se van a utilizar en el desarrollo del juego, y como en el resto de pantallas de la aplicación tiene la opción de poder silenciar o no el sonido.



Imagen 36. Ayuda